Group Signatures with Message-Dependent Opening: Formal Definitions and Constructions

著者（英）	Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, Kazuma Ohara, Kazumasa OMOTE, Yusuke Sakai
journal or publication title	Security and communication networks
volume	2019
page range	4872403
year	2019-08
権利	(C) 2019 Keita Emura et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
URL	http://hdl.handle.net/2241/00159453

doi: 10.1155/2019/4872403

Hindawi
Security and Communication Networks
Volume 2019, Article ID 4872403, 36 pages
https://doi.org/10.1155/2019/4872403

Research Article

Group Signatures with Message-Dependent Opening: Formal Definitions and Constructions

Keita Emura,[1] **Goichiro Hanaoka**,[2] **Yutaka Kawai**,[3] **Takahiro Matsuda**,[2] **Kazuma Ohara**,[4] **Kazumasa Omote**,[5] **and Yusuke Sakai**[2]

[1]*National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan*
[2]*National Institute of Advanced Industrial Science and Technology, 2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan*
[3]*Mitsubishi Electric, 5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japan*
[4]*The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan*
[5]*University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan*

Correspondence should be addressed to Yusuke Sakai; yusuke.sakai@aist.go.jp

Received 5 April 2019; Accepted 7 July 2019; Published 26 August 2019

Academic Editor: David Megias

This paper introduces a new capability for group signatures called *message-dependent opening*. It is intended to weaken the high trust placed on the opener; i.e., no anonymity against the opener is provided by an ordinary group signature scheme. In a group signature scheme with message-dependent opening (GS-MDO), in addition to the opener, we set up an *admitter* that is not able to extract any user's identity but *admits* the opener to open signatures by specifying messages where signatures on the specified messages will be opened by the opener. The opener cannot extract the signer's identity from any signature whose corresponding message is not specified by the admitter. This paper presents formal definitions of GS-MDO and proposes a generic construction of it from identity-based encryption and adaptive non-interactive zero-knowledge proofs. Moreover, we propose two specific constructions, one in the standard model and one in the random oracle model. Our scheme in the standard model is an instantiation of our generic construction but the message-dependent opening property is bounded. In contrast, our scheme in the random oracle model is not a direct instantiation of our generic construction but is optimized to increase efficiency and achieves the unbounded message-dependent opening property. Furthermore, we also demonstrate that GS-MDO implies identity-based encryption, thus implying that identity-based encryption is essential for designing GS-MDO schemes.

1. Introduction

Group signatures [1] are anonymous signatures that allow members of a group to anonymously sign messages on behalf of the group. Signatures are verified with a single group public key, and the verification process does not reveal the identity of the signer. A designated authority, called the opener, identifies the actual signer in various exceptional cases. However, ordinary group signatures provide the opener with an extreme privilege; i.e., the opener can freely identify the originator of any signature that he chooses. In other words, ordinary group signature schemes provide absolutely no assurance of privacy against the opener. For example, in an anonymous auction the opener can extract all bidders' identities, which will be explained later in more detail.

This paper investigates a way of "decentralizing" this strong authority of the opener. Towards this end, we propose a new kind of group signatures involving *the message-dependent opening (MDO) property*. It divides (or decentralizes) the strong authority of the opener by introducing another authority called the *admitter*. In exceptional cases that a signature on a problematic message is found, the admitter issues a *token* that corresponds to the message (as opposed to all signed messages). The opener extracts the signer's identity from the signature using this token, whereas without it, he is not able to do so. For instance, in an anonymous bulletin board system using our group signature scheme, if the admitter decides that the message "Mr. XXX is a fool!" should not be publicized as a signed message by an anonymous group member, he issues a token for this message.

Then, by using it, the opener can immediately identify the signer's identity of any signature if it corresponds to this message.

At first glance, one may think that the popular thresholding technique (i.e., thresholding the opener into multiple less-trusted openers) [2] would already be sufficient to achieve the above property. However, this is not true. Namely, in our context, the token is generated on *the message that the admitter chooses* but not the signature for such messages. Therefore, once a token for a message (which is chosen by the admitter) is issued, the signers' identities for all signatures on this message can immediately be extracted by the opener without him having to interact with any other party. Consequently, the opener can noninteractively identify the signer's identity for a message that has already been specified as problematic. Furthermore if the admitter considers that there is no longer any need to specify further messages that should be opened, then he can erase his memory to avoid having his secret leaked. Note that even when the admitter has erased his secret, the opener can still open the signer's identity of any signature provided that its corresponding message was previously specified by the admitter.

1.1. Contributions. This paper proposes group signatures with a new additional capability, called *group signatures with message-dependent opening* (GS-MDO). We introduce an *admitter* in GS-MDO, as previously mentioned, that issues tokens for specific messages, and by using these tokens, the opener can extract signers' identities from signatures only if their corresponding messages have been specified. We can flexibly restrict the capabilities of the opener utilizing this property without implementing any complicated interactive procedures (e.g., threshold decryption).

The main contributions of this paper are threefold. First, we provide a formal model and a security definition for GS-MDO. Our model and security definition are extensions of the Bellare-Micciancio-Warinschi (BMW) model [3], which is considered to be the basic security definition for group signatures in the static setting. More specifically, our security model is a natural modification of this model extended according to the difference between a standard group signature scheme and ours which introduces the MDO property. In addition, we demonstrate that *it is possible to derive identity-based encryption (IBE) from any GS-MDO scheme in a black-box manner* if the underlying GS-MDO is secure in the above sense.

Secondly, we present a generic construction of GS-MDO from a public key encryption (PKE) scheme, an IBE scheme, and an adaptive noninteractive zero-knowledge (NIZK) proof system. (Technically, we use a tag-based KEM (key encapsulation mechanism) and an identity-based KEM instead of a PKE scheme and an IBE scheme). Given the above result that GS-MDO implies IBE, it is expected to be quite hard to construct GS-MDO without using IBE or similar primitives as a building block. We note that simulation-soundness [4] is not required for NIZK in our generic construction, while the generic construction of the (ordinary) group signature [3] requires this strong property.

Finally, we propose two efficient instantiations of GS-MDO, one in the standard model and one in the random oracle model. Our scheme in the standard model uses the Groth-Sahai proof system [5] as an NIZK proof system in our generic construction. To utilize the Groth-Sahai proof system in our generic construction, we note that an IBE scheme that is compatible with the Groth-Sahai proof (such as structure-preserving signatures [6]) is necessary since our generic construction requires IBE. To be compatible with the Groth-Sahai proof system, an IBE scheme needs to be able to encrypt a base-group element and to have its ciphertext consist of only base-group elements. Although Libert and Joye [7] proposed an IBE scheme whose plaintext space is a base group, and the IBE scheme is compatible with the Groth-Sahai proof, the size of the group signature of the resulting GS-MDO scheme depends on the number of group members. In order to construct a GS-MDO scheme with constant-size signature, we also construct a new IBE scheme that has only k-resilient security [8]. Therefore, the resulting GS-MDO scheme inherits this restriction (i.e., the admitter can issue at most k tokens). The size of a signature is approximately 10 kilobytes when a 170-bit prime-order group is used. (We adopt the estimates of bit sizes for group elements used in [9]). Our scheme in the random oracle model, on the other hand, is based on the Boneh-Boyen-Shacham (BBS) group signature [10] and uses the Boneh-Franklin (BF) IBE scheme [11] and a variant of the Cramer-Shoup encryption scheme [12, 13]. As we can use IBE (which is not k-resilient) in the random oracle model, GS-MDO with an unbounded MDO property (i.e., the number of tokens issued by the admitter is unlimited) can thus be constructed. The idea behind this construction is based on our generic construction, but the instantiation is not straightforward since the PKE scheme and the IBE scheme cannot directly be combined. Hence, we modify the building blocks and the construction to combine them. To be more precise, we design a dedicated extension of the linear encryption scheme [10] with chosen-ciphertext security and public verifiability to enable our efficient construction. The size of the signature of the scheme is 3636 bits in 80-bit security, which is 20 times smaller than that of our standard model scheme.

Note that our security notions and proposed schemes assume that the system setup is carried out honestly. This setup includes the generation of a common reference string, the opener's and the admitter's secret keys, and the group members' signing keys. However, the goal of the current paper is to provide a feasibility study of the MDO property in group signatures. Namely, we study the possibility of constructing a GS-MDO scheme in a simple setting where several parameters and secret keys are generated honestly. We defer to future work a more comprehensive study on the possibility of constructing a scheme with a more complicated and realistic scenario such as that allowing maliciously generated parameters and on providing a hedge against this types of attack.

1.2. Applications. As previously mentioned, a straightforward application of GS-MDO schemes is in detecting the

originators of inappropriate messages in anonymous bulletin board systems. We further discuss more potential applications of GS-MDO schemes in what follows.

The first application we discuss is *anonymous auctions*, where bidders form a group of anonymous signers. Each bidder produces a group signature on his bidding price. The admitter issues a token for opening signatures on the highest price to detect the winner(s). The opener is then only able to open the signatures on the highest price.

The main advantage (of the MDO approach) over the threshold approach becomes clear in this application. Suppose that there are *many* winners who all have bid the highest price resulting in a tie. As an interaction will be needed for each winner in the threshold approach, the total communication cost will be proportional to the number of winners. In contrast, if one takes the MDO approach, only a small communication bandwidth from the admitter to the opener is needed. The communication cost does not depend on the number of winners.

Another application in which the MDO property is useful is *identity escrow*. Although our primitive is a rather general purpose primitive for identity escrow scenarios, for the ease of understanding, let us discuss this with a concrete example. Consider a customer who enters an automated parking garage [14], where he generates a group signature on a message that encodes the date when he enters the car park (say, the string "2012-02-20"). Let us assume a felony has been committed (e.g., a person is murdered) in the garage. The opener in this case will open the signatures on the date when the murder occurred to identify who was there on that day.

The opener in this application needs to open *many* signatures on the *same* message. If one adopts the threshold technique to decentralize the authority, many interactions are required to open all the signatures. The MDO property removes interactions between authorities, i.e., the admitter issues a token for the day, and the opener opens all the signatures without interactions.

As an application of GS-MDO, Arai et al. [15] proposed a privacy-preserving anomaly detection framework. In their framework, the admitter plays the role of analyzing data, and the opener plays the role of detecting adversarial users. Briefly, a user generates a group signature on data, and sends it to the admitter. Then, the admitter can detect anomaly data without identifying users. For detecting adversarial users who generate anomaly data, the admitter generates tokens that correspond to the anomaly data, and sends them to the opener. This framework allows the admitter to detect adversarial users, while other honest users are kept anonymous.

1.3. Related Work. Since group signatures were first proposed by Chaum and van Heyst [1], many efficient constructions have been proposed, most of which depend on a random oracle [10, 16–23]. Many earlier schemes were based on the strong RSA assumption [24, 25]. Group signature schemes based on assumptions related to the discrete-logarithm type were achieved, to name a few, by Camenisch and Lysyanskaya [17] and by Boneh, Boyen, and Shacham [10]. The former scheme is based on the LRSW assumption [26], while the latter is based on the q-strong Diffie-Hellman (q-SDH) assumption [27].

Group signature schemes without random oracles were also achieved. Ateniese et al. [28] first proposed a group signature scheme from interactive assumptions avoiding random oracles. Following this scheme, Groth proposed a group signature scheme that avoids random oracles and interactive assumptions [29], but his scheme has a very large signature size. Boyen and Waters [30, 31] proposed highly efficient constructions, although the security guarantees of their schemes are not very strong, i.e., they only achieve so-called CPA-anonymity. Groth [32] proposed another group signature scheme, which is almost as efficient as the Boyen-Waters schemes and satisfies higher security guarantee of the Bellare-Shi-Zhang (BSZ) model [33]. Libert et al. [34] proposed a group signature scheme secure in the standard model from standard assumptions.

Regarding decentralizing and distributing the power of the group manager, the separability of a cryptographic protocol was introduced by Kilian and Petrank [14] in the context of identity escrow. Later, this notion was refined and adopted to the context of group signature by Camenisch and Michels [35]. The separability notion requires that keys of several entities involved in a cryptographic primitive be generated independently of each other. In their setting, the power of a group manager is separated into two authorities. The first authority is able to allow a new member to join the group but is not able to identify the originator of a group signature, and the other authority is able to identify the originator of a group signature but is not able to allow a new member to join the group. More formal modeling of these separated authorities has been proposed by Bellare et al. [33] and Kiayias and Yung [36]. Sakai et al. [37] further extended the BSZ model by considering signature hijacking attacks.

Libert et al. [38, 39] proposed scalable group signature schemes with revocation, and Attrapadung et al. [40] and Nakanishi et al. [41] also proposed revocable group signature schemes with a compact revocation list.

Traceable signatures are an extended notion of group signatures and were introduced by Kiayias, Tsiounis, and Yung [42]. This primitive allows the group manager to specify a group member as "misbehaving". Once a member is specified by the manager, anyone becomes able to detect the signatures of the specified user without interacting with the manager. In this case, signatures of other group members continue to be anonymous. In our terminology, this primitive achieves a somewhat "signer-dependent opening" property, but the MDO property is not achieved. A contractual anonymity system [43] based on group signatures with verifier-local revocation [44] has been proposed. In this system, when a user breaks a contract, an accountability server revokes the anonymity of the user and notifies the identity of the user to the service provider. (In this system, a user is said to *break a contract* when the user sends a message breaking the contract policy of the service provider). Since this scheme uses a conventional open algorithm, this system also differs from MDO.

As follow-up works to our results [45, 46], Libert and Joye [7] proposed an unbounded GS-MDO scheme in the standard model with logarithmic signature size. They proposed a partially structure-preserving IBE scheme and used it as a building block of GS-MDO. The signature of their scheme consists of $53 \log N + 35$ group elements, where N is the number of group members, whereas our GS-MDO schemes achieve constant-size signatures, though our standard model scheme does not achieve the unbounded MDO property. Constructing an unbounded GS-MDO scheme secure in the standard model with constant-size signatures is an interesting open problem. Libert, Mouhartem, and Nguyen [47] proposed a lattice-based (unbounded) GS-MDO scheme in the random oracle model. Their scheme was proved secure from the short integer solution (SIS) assumption and the learning with errors (LWE) assumption. The signature size of this scheme is still logarithmic in the number of signers, whereas our two constructions have constant-size signatures.

Preliminary versions of this paper were presented at the 5th International Conference on Pairing-Based Cryptography (Pairing 2012) [45] and at the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2013) [46]. This is the merged full version. In this version, we give security proofs omitted in the proceedings versions.

1.4. Paper Organization. The rest of this paper is structured as follows. Section 2 briefly describes definitions and security notions of several building blocks. Section 3 presents the notion of GS-MDO, its syntax and security definitions. Section 4 discusses difficulties behind constructing efficient GS-MDO schemes. Specifically, we argue the use of IBE in a construction of GS-MDO is essential by showing a generic construction of IBE from GS-MDO. In Section 5, we propose a generic construction of GS-MDO, and Sections 6 and 7 show reasonably efficient instantiations in the standard model and the random oracle model, respectively. In Section 8, we compare the efficiency of our GS-MDO schemes with previous group signatures. In Section 9, we conclude the paper and list open problems.

2. Preliminaries

In this section, we describe the syntax of building blocks and some computational assumptions we use for constructing GS-MDO. Throughout the paper, we use the following notations: $x \longleftarrow X$ denotes that x is uniformly and independently sampled from the set X. For an algorithm $\mathcal{A}$, an input x, and a randomness r we denote $y \longleftarrow \mathcal{A}(x; r)$ to run $\mathcal{A}$ with an input x and a randomness r and let y be the output. When the randomness is implicit, we denote $y \longleftarrow \mathcal{A}(x)$. For an integer $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$.

2.1. Signatures. A signature scheme consists of the following three algorithms:

(i) SigKg: this is the key generation algorithm that takes as an input a security parameter 1^λ and outputs a pair (vk, sk) of a verification key and a signing key.

(ii) Sign: this is the signing algorithm that takes as inputs a signing key sk and a message M and outputs a signature s on the message M.

(iii) Verify: this is the verification algorithm that takes as inputs a verification key vk, a message M, and a signature s and outputs $\top$ or $\bot$, which, respectively, indicate "accept" or "reject."

For correctness, we require that for all $\lambda \in \mathbb{N}$, all pairs $(vk, sk) \longleftarrow \mathsf{SigKg}(1^\lambda)$, and all messages M, it is satisfied that $\mathsf{Verify}(vk, M, \mathsf{Sign}(sk, M)) = \top$.

A signature scheme is said to be *existentially unforgeable under chosen-message attacks (EUF-CMA)*, if for any probabilistic polynomial-time adversary the advantage in the following game is negligible:

(i) *Setup*: the challenger runs $(vk, sk) \longleftarrow \mathsf{SigKg}(1^\lambda)$ and gives the adversary vk.

(ii) *Query*: the adversary adaptively issues signing queries M in an arbitrary order. For each signing query M, the challenger runs $s \longleftarrow \mathsf{Sign}(sk, M)$ and returns s to the adversary.

(iii) *Forge*: finally the adversary outputs a forgery (M^*, s^*). The adversary wins the game if M^* is not queried as a signing query, and $\mathsf{Verify}(vk, M^*, s^*) = \top$.

The advantage of the adversary $\mathcal{A}$ is defined as the probability that the adversary wins and is denoted by $\mathrm{Adv}^{\text{EUF-CMA}}_{\mathcal{A}}(\lambda)$.

A signature scheme is said to be *a strongly unforgeable one-time signature scheme*, if for any probabilistic polynomial-time adversary the advantage in the following game is negligible:

(i) *Setup*: the challenger runs $(vk, sk) \longleftarrow \mathsf{SigKg}(1^\lambda)$ and gives the adversary vk.

(ii) *Query*: the adversary issues a signing query M *only once*. The challenger runs $s \longleftarrow \mathsf{Sign}(sk, M)$ and returns s to the adversary.

(iii) *Forge*: finally the adversary outputs a forgery (M^*, s^*). The adversary wins the game if $(M^*, s^*) \neq (M, s)$ and $\mathsf{Verify}(vk, M^*, s^*) = \top$.

The advantage of the adversary $\mathcal{A}$ is defined as the probability that the adversary wins and is denoted by $\mathrm{Adv}^{\text{OT-CMA}}_{\mathcal{A}}(\lambda)$.

2.2. Tag-Based Key Encapsulation Mechanism. A tag-based key encapsulation mechanism (tag-based KEM) (Tag-based encryption, an encryption analogue of tag-based KEM, is originally introduced as "encryption with labels" by Shoup and Gennaro [49]. Tag-based KEM is different from "tag-KEM", introduced by Abe, Gennaro, Kurosawa, and Shoup [50]. However, any chosen-ciphertext secure tag-KEM can be immediately converted to a tag-based KEM satisfying security that is sufficient for our purpose) [51, 52] scheme consists of the following three algorithms:

(i) TKg: this is the key generation algorithm that takes as an input a security parameter 1^λ and outputs a pair (pk, dk) of a public key and a secret key.

(ii) TEnc: this is the encapsulation algorithm that takes as inputs a public key pk and a tag t and outputs (C, K) where a ciphertext C for a tag t encapsulates a session key $K \in \mathcal{K}_{\mathrm{PKE}}$ and $\mathcal{K}_{\mathrm{PKE}}$ is the session key space associated with the scheme.

(iii) TDec: this is the deterministic decapsulation algorithm that takes as inputs a secret key dk, a tag t, and a ciphertext C and outputs a decapsulated session key K or a special symbol $\perp$ indicating an invalid ciphertext.

For correctness, we require that for all $\lambda \in \mathbb{N}$, all $(pk, dk) \longleftarrow \mathsf{TKg}(1^\lambda)$, all tags $t \in \{0,1\}^*$, and all $(C, K) \longleftarrow \mathsf{TEnc}(pk, t)$, it holds that $\mathsf{TDec}(dk, t, C) = K$.

A tag-based KEM is said to be *selective-tag weakly chosen-ciphertext secure* if for any probabilistic polynomial-time adversary the advantage in the following game is negligible:

(i) *Setup*: the adversary is given a security parameter 1^λ and outputs a challenge tag t^*. The challenger runs $(pk, dk) \longleftarrow \mathsf{TKg}(1^\lambda)$, then the challenger gives the adversary the public key pk.

(ii) *Query (Phase I)*: the adversary issues decryption queries (t, C) in an arbitrary order. The challenger runs $K \longleftarrow \mathsf{TDec}(dk, t, C)$ and returns K to the adversary. Here the adversary is not allowed to issue queries with $t = t^*$.

(iii) *Challenge*: at some point the adversary requests a challenge. The challenger chooses a random bit $b \longleftarrow \{0,1\}$, runs $(K_0, C^*) \longleftarrow \mathsf{TEnc}(pk, t^*)$, chooses $K_1 \longleftarrow \mathcal{K}_{\mathrm{PKE}}$, and sends (C^*, K_b).

(iv) *Query (Phase II)*: after receiving the challenge the adversary is again allowed to issue decryption queries. The same restriction for queries is applied as before.

(v) *Guess*: finally the adversary outputs a bit b'. The adversary wins the game if $b = b'$.

The advantage of the adversary $\mathcal{A}$ is defined by $|\Pr[b = b'] - 1/2|$ and is denoted by $\mathrm{Adv}_{\mathcal{A}}^{\text{tb-KEM}}(\lambda)$.

2.3. Identity-Based KEM and Its k-Resilient Variant. An identity-based KEM [8] consists of the following four algorithms:

(i) ISetup: this is the setup algorithm that takes as inputs a security parameter 1^λ and a collusion threshold 1^k and outputs a pair (par, mk) of a public parameter and a master secret key.

(ii) IExt: this is the key extraction algorithm that takes as inputs a master secret key mk and an identity ID and outputs a user decapsulation key dk_{ID}.

(iii) IEnc: this is the encapsulation algorithm that takes as inputs a public parameter par and an identity ID and outputs (C, K) where a ciphertext C for an identity ID encapsulates a session key $K \in \mathcal{K}_{\mathrm{IBE}}$ and $\mathcal{K}_{\mathrm{IBE}}$ is the session key space associated with the scheme.

(iv) IDec: this is the deterministic decapsulation algorithm that takes as inputs dk_{ID}, ID, and C and outputs a decapsulated session key K or a special symbol $\perp$ indicating an invalid ciphertext.

For correctness, we require that for all $\lambda \in \mathbb{N}$, all $k \in \mathbb{N}$, all $(par, mk) \longleftarrow \mathsf{ISetup}(1^\lambda, 1^k)$, all identities $ID \in \{0,1\}^*$, all $dk_{ID} \longleftarrow \mathsf{IExt}(mk, ID)$, and all $(C, K) \longleftarrow \mathsf{IEnc}(par, ID)$, it holds that $\mathsf{IDec}(dk_{ID}, ID, C) = K$.

An identity-based KEM is said to be *k-resilient* if for any probabilistic polynomial-time adversary the advantage in the following game is negligible:

(i) *Setup*: the challenger runs $(par, mk) \longleftarrow \mathsf{ISetup}(1^\lambda)$, then the challenger gives the adversary the public parameter par.

(ii) *Query (Phase I)*: the adversary issues extraction queries ID in an arbitrary order. The challenger runs $dk_{ID} \longleftarrow \mathsf{IExt}(mk, ID)$ and returns dk_{ID} to the adversary. The total number (the summation of those in Phase I and Phase II) of extraction queries during the game is restricted to be smaller than or equal to k.

(iii) *Challenge*: at some point the adversary requests a challenge with an identity ID^*. The challenger chooses a random bit $b \longleftarrow \{0,1\}$, runs $(C^*, K_0) \longleftarrow \mathsf{IEnc}(par, ID^*)$, chooses $K_1 \longleftarrow \mathcal{K}_{\mathrm{IBE}}$, and sends (C^*, K_b). The adversary is not allowed to request a challenge with an identity whose user decapsulation key is queried before.

(iv) *Query (Phase II)*: after receiving the challenge the adversary is again allowed to issue extraction queries. This time querying a user decapsulation key for ID^* is disallowed.

(v) *Guess*: finally the adversary outputs a bit b'. The adversary wins the game if $b = b'$.

The advantage of the adversary $\mathcal{A}$ is defined by $|\Pr[b = b'] - 1/2|$ and is denoted by $\mathrm{Adv}_{\mathcal{A}}^{\text{ib-KEM}}(\lambda)$.

A *fully secure* IBE scheme is defined analogously with the difference that the setup algorithm does not take 1^k as an input and in the game the number of extraction queries is unbounded.

2.4. Adaptive Noninteractive Zero-Knowledge Proofs. A noninteractive proof system for a polynomial-time computable relation R consists of the following three algorithms:

(i) K: this is the common reference string generation algorithm that takes as an input a security parameter 1^λ and outputs a common reference string Σ.

(ii) P: this is the proof algorithm that takes as inputs a common reference string Σ, a statement x, and a witness w, where $R(x, w) = \top$ and outputs a proof π.

(iii) V: this is the verification algorithm that takes as inputs a common reference string Σ, a statement x, and a proof π and outputs either $\top$ or $\perp$.

We say that a noninteractive proof system (K, P, V) has perfect completeness, if for all $\lambda \in \mathbb{N}$, for all (x, w) such that

$R(x, w) = \top$, for all $\Sigma \longleftarrow K(1^\lambda)$, for all $\pi \longleftarrow P(\Sigma, x, w)$ it holds that $V(\Sigma, x, \pi) = \top$.

A noninteractive proof system is said to have adaptive perfect soundness, if for any probabilistic polynomial-time adversary the advantage in the following game is equal to zero:

(i) *Setup*: the challenger runs $\Sigma \longleftarrow K(1^\lambda)$ and gives the adversary the common reference string Σ.

(ii) *Forge*: the adversary outputs a pair (x, π) of a statement and a proof. The adversary wins if $V(\Sigma, x, \pi) = \top$ and $R(x, w) = \perp$ for any w.

The advantage of the adversary $\mathscr{A}$ is defined by the probability that the adversary wins and is denoted by $\mathrm{Adv}^{\mathrm{sound}}_{\mathscr{A}}(\lambda)$.

We say that a noninteractive proof system (K, P, V) is adaptively zero-knowledge if there is a pair of algorithms (S_1, S_2) such that for any probabilistic polynomial-time adversary the advantage in the following game is negligible:

(i) *Setup*: the challenger generates a bit $b \longleftarrow \{0, 1\}$. If $b = 0$ the challenger runs $\Sigma \longleftarrow K(1^\lambda)$. If $b = 1$ the challenger runs $(\Sigma, \tau) \longleftarrow S_1(1^\lambda)$. Then the challenger gives the adversary the common reference string Σ.

(ii) *Query*: the adversary is allowed to issue a query (x, w) such that $R(x, w) = \top$ *only once*. If $b = 0$ the challenger runs $\pi \longleftarrow P(\Sigma, x, w)$. If $b = 1$ the challenger runs $\pi \longleftarrow S_2(\Sigma, \tau, x)$. The challenger gives the adversary π.

(iii) *Guess*: finally the adversary outputs a bit b'.

The advantage of the adversary $\mathscr{A}$ is defined by $|\Pr[b = b'] - 1/2|$ and is denoted by $\mathrm{Adv}^{\mathrm{zk}}_{\mathscr{A}}(\lambda)$.

2.5. Computational Assumptions. Let $\mathscr{G}$ be a probabilistic polynomial-time algorithm that takes a security parameter 1^λ as an input and generates a parameter $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ of *bilinear groups*, where p is a λ-bit prime, $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order p, e is a bilinear map from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$, and g is a generator of $\mathbb{G}$. We then describe several computational assumptions on which the proposed schemes are based.

The q-Strong Diffie-Hellman Assumption [27]. Let $q \in \mathbb{N}$ and let $(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathscr{G}(1^\lambda)$, $\gamma \longleftarrow \mathbb{Z}_p$ and $A_i \longleftarrow g^{\gamma^i}$ for $i \in [q]$. The q-strong Diffie-Hellman problem in $\mathbb{G}$ is stated as follows: given $(g, (A_i)_{i\in[q]})$, output $(c, g^{1/(\gamma+c)})$ where $c \in \mathbb{Z}_p$. The advantage of an algorithm $\mathscr{A}$ in solving the q-strong Diffie-Hellman problem is defined as

$$\mathrm{Adv}^{\text{q-SDH}}_{\mathscr{A}}(\lambda) = \Pr\left[\mathscr{A}\left(g, (A_i)_{i\in[q]}\right) = \left(c, g^{1/(\gamma+c)}\right) \wedge c \in \mathbb{Z}_p\right]. \tag{1}$$

We say that the q-strong Diffie-Hellman (q-SDH) assumption holds if $\mathrm{Adv}^{q\text{-SDH}}_{\mathscr{A}}(\lambda)$ is negligible in λ for any probabilistic polynomial-time algorithm $\mathscr{A}$.

For some fixed base $g \in \mathbb{G}$ and a fixed group element $w = g^\gamma \in \mathbb{G}$ we say that a pair $(x, A) \in \mathbb{Z}_p \times \mathbb{G}$ is an *SDH pair* if A is equal to $g^{1/(\gamma+x)}$.

The Decision Linear Assumption [10]. Let $(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathscr{G}(1^\lambda)$, $u, v, h \longleftarrow \mathbb{G} \setminus \{1\}$, $\alpha, \beta \longleftarrow \mathbb{Z}_p$, and $\gamma \longleftarrow \mathbb{Z}_p \setminus \{\alpha + \beta\}$. The decision linear problem in $\mathbb{G}$ is stated as follows: distinguish the distribution $(u, v, h, u^\alpha, v^\beta, h^{\alpha+\beta})$ from the distribution $(u, v, h, u^\alpha, v^\beta, h^\gamma)$. The advantage of an algorithm $\mathscr{A}$ in solving the decision linear problem is defined as

$$\mathrm{Adv}^{\mathrm{DLIN}}_{\mathscr{A}}(\lambda) = \left|\Pr\left[\mathscr{A}\left(u, v, h, u^\alpha, v^\beta, h^{\alpha+\beta}\right) = 1\right] - \Pr\left[\mathscr{A}\left(u, v, h, u^\alpha, v^\beta, h^\gamma\right) = 1\right]\right|. \tag{2}$$

We say that the decision linear (DLIN) assumption holds if $\mathrm{Adv}^{\mathrm{DLIN}}_{\mathscr{A}}(\lambda)$ is negligible in λ for any probabilistic polynomial-time algorithm $\mathscr{A}$.

For some fixed bases $u, v, h \in \mathbb{G}$, we say that a tuple (T_1, T_2, T_3) is a *linear tuple* if there exist exponents $\alpha, \beta \in \mathbb{Z}_p$ that satisfy $(T_1, T_2, T_3) = (u^\alpha, v^\beta, h^{\alpha+\beta})$.

The Decision Bilinear Diffie-Hellman Assumption [53]. Let $(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathscr{G}(1^\lambda)$ and $a, b, c, d \longleftarrow \mathbb{Z}_p$. The decision bilinear Diffie-Hellman problem in $(\mathbb{G}, \mathbb{G}_T)$ is stated as follows: distinguish the distribution $(g, g^a, g^b, g^c, e(g, g)^{abc})$ from the distribution $(g, g^a, g^b, g^c, e(g, g)^d)$. The advantage of an algorithm $\mathscr{A}$ in solving the decision bilinear Diffie-Hellman problem is defined as

$$\mathrm{Adv}^{\mathrm{DBDH}}_{\mathscr{A}}(\lambda) = \left|\Pr\left[\mathscr{A}\left(g, g^a, g^b, g^c, e\left(g, g\right)^{abc}\right) = 1\right] - \Pr\left[\mathscr{A}\left(g, g^a, g^b, g^c, e\left(g, g\right)^d\right) = 1\right]\right|. \tag{3}$$

We say that the decision bilinear Diffie-Hellman (DBDH) assumption holds if $\mathrm{Adv}^{\mathrm{DBDH}}_{\mathscr{A}}(\lambda)$ is negligible in λ for any probabilistic polynomial-time algorithm $\mathscr{A}$.

The Simultaneous Flexible Pairing Assumption [54]. Let $(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathscr{G}(1^\lambda)$, $g_z, g_r, h_z, h_u \longleftarrow \mathbb{G}$ are generators of $\mathbb{G}$ and $(a, \tilde{a}), (b, \tilde{b}) \longleftarrow \mathbb{G} \times \mathbb{G}$. For $i \in [q]$, let $R_i = (z_i, r_i, s_i, t_i, u_i, v_i, w_i) \in \mathbb{G}^7$ which satisfies $e(a, \tilde{a}) = e(g_z, z_i)e(g_r, r_i)e(s_i, t_i)$ and $e(b, \tilde{b}) = e(h_z, z_i)e(h_u, u_i)e(v_i, w_i)$. The simultaneous flexible pairing problem is stated as follows: given $(g_z, g_r, h_z, h_u, a, \tilde{a}, b, \tilde{b})$ and $R_1, \ldots, R_q$, output $R^* = (z^*, r^*, s^*, t^*, u^*, v^*, w^*)$ where $e(a, \tilde{a}) = e(g_z, z^*)e(g_r, r^*)e(s^*, t^*)$ and $e(b, \tilde{b}) = e(h_z, z^*)e(h_u, u^*)e(v^*, w^*)$ satisfying $z^* \neq 1$ and $z^* \neq z_i$ for every i. The advantage of an algorithm $\mathscr{A}$ in solving the simultaneous flexible pairing problem is defined as

$$\mathrm{Adv}^{\mathrm{SFP}}_{\mathscr{A}}(\lambda) = \Pr\left[\mathscr{A}\left(g_z, g_r, h_z, h_u, a, \tilde{a}, b, \tilde{b}, (R_i)_{i\in[q]}\right) = (z^*, r^*, s^*, t^*, u^*, v^*, w^*) \wedge e(a, \tilde{a})\right.$$

$$= e(g_z, z^*)\, e(g_r, r^*)\, e(s^*, t^*) \wedge e\left(b, \tilde{b}\right)$$
$$= e(h_z, z^*)\, e(h_u, u^*)\, e(v^*, w^*) \wedge z^* \neq 1 \wedge z^* \neq z_i \ \forall i\Big]. \quad (4)$$

We say that the simultaneous flexible pairing (SFP) assumption holds if $\mathrm{Adv}^{\mathrm{SFP}}_{\mathscr{A}}(\lambda)$ is negligible in λ for any probabilistic polynomial-time algorithm $\mathscr{A}$.

3. Group Signatures with Message-Dependent Opening

In this section we introduce the syntax and security definitions for GS-MDO. As an ordinary group signature scheme, a GS-MDO scheme allows group members to sign a message anonymously, that is, without revealing their identities but only showing that one of the group members actually signed. In exceptional cases, a designated third party, called the opener, can "open" exceptional signatures to identify the originator of the signatures. In contrast to ordinary group signature schemes, a GS-MDO scheme requires the opener to cooperate with another authority, called the admitter, to open signatures. The admitter issues a message-specific token, and the opener is able to open a signature on the message *only when a token for the message is issued from the admitter*.

In this section, we define the formal model and the security requirements of GS-MDO.

3.1. The Model of GS-MDO. Formally, a GS-MDO scheme consists of the following five probabilistic polynomial-time algorithms:

(i) GKg: this is the key generation algorithm that takes as inputs a security parameter 1^λ, the number of group members 1^n, and the maximum number 1^k of message-specific tokens that can be issued and outputs a group public key gpk, an admitting key ak, an opening key ok, and n group signing keys $(gsk_i)_{i\in[n]}$. We note that the input 1^k is omitted when the number of message-specific tokens that can be issued is unbounded.

(ii) GSig: this is the signing algorithm that takes as inputs a group public key gpk, a group signing key gsk_i, and a message M and outputs a group signature σ.

(iii) Td: this is the message-specific token generation algorithm that takes as inputs a group public key gpk, an admitting key ak, and a message M and outputs a token t_M for M.

(iv) GVf: this is the verification algorithm that takes as inputs a group public key gpk, a message M, and a group signature σ and outputs $\top$ (that means accept) or $\perp$ (that means reject).

(v) Open: this is the opening algorithm that takes as inputs a group public key gpk, an opening key ok, a message M, a group signature σ, and a message-specific token t_M and outputs a user index $i \in \mathbb{N}$ or $\perp$.

For correctness, we require that for all λ, n, $k \in \mathbb{N}$, for all $(gpk, ak, ok, (gsk_i)_{i\in[n]}) \longleftarrow \mathsf{GKg}(1^\lambda, 1^n, 1^k)$, for all messages $M \in \{0,1\}^*$ and all user $i \in [n]$, it holds that $\mathsf{GVf}(gpk, M, \mathsf{GSig}(gpk, gsk_i, M)) = \top$ and $\mathsf{Open}(gpk, ok, M, \mathsf{GSig}(gpk, gsk_i, M), \mathsf{Td}(gpk, ak, M)) = i$.

In the above scenario, the signing keys $(gsk_i)_{i\in[n]}$, the opening key ok, and the admitting key ak are generated by the GKg algorithm in the setup phase. Then, ok is given to the opener and ak is given to the admitter. A signer can make a group signature σ by using the GSig algorithm, and all entities can verify the group signature by the GVf algorithm and public key gpk. This setting is the same as ordinary group signatures, except the group manager's key is divided into ok and ak. In exceptional cases, the admitter issues a message-specific token t_M by using the Td algorithm. Moreover, the opener can identify the signer of a group signature by using the Open algorithm only when the opener received t_M from the admitter.

3.2. Security Requirements. We introduce the security notion for GS-MDO. In contract to ordinary group signatures, we need to take care of the MDO property.

Ordinary group signatures are required to ensure two security notions, anonymity and traceability, whereas we have to further ensure two types of anonymity in the case of GS-MDO that are related to the original motivation for the introduction of the admitter. The introduction of the admitter is intended to strengthen signers' anonymity against the authorities as much as possible. To capture this intention, we define the indistinguishability of the originator of a signature in the strong setting where the opening key is given to the adversary (opener anonymity). As a counterpart of this, we also define indistinguishability in the setting where the admitting key is given to the adversary (admitter anonymity). Note that we do not consider the situation in which the adversary obtains both the opening key and the admitting key because in this situation the adversary can open any signature by itself.

For traceability, we use the same definition as for the ordinary group signatures, in which the authorities are entirely corrupted by the adversary, since even ordinary group signature schemes ensure this level of traceability against entirely corrupted openers.

3.2.1. Opener Anonymity. Here we give the formal definition of anonymity against the opener, which we call *opener anonymity*. It is formalized as the indistinguishability of signatures of two different signers of the adversary's choice. Opener anonymity is defined by requiring that no adversary has nonnegligible advantage in distinguishing signatures. We again remark that contrary to ordinary group signatures, the adversary is allowed to have the opening key. This is intended for modeling "anonymity against the opener."

Definition 1. A GS-MDO scheme is said to have *opener anonymity* if for any probabilistic polynomial-time adversary $\mathcal{A}$ the advantage in the following game is negligible:

(i) *Setup*: the challenger runs $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ to obtain $(gpk, ok, ak, (gsk_i)_{i\in[n]})$. The challenger sends $(gpk, ok, (gsk_i)_{i\in[n]})$ to $\mathcal{A}$.

(ii) *Token Query (Phase I)*: $\mathcal{A}$ adaptively issues token queries M in an arbitrary order. The challenger runs $t_M \longleftarrow \mathsf{Td}(gpk, ak, M)$ and returns t_M to $\mathcal{A}$.

(iii) *Challenge*: at some point $\mathcal{A}$ requests a challenge for i_0, $i_1 \in [n]$, and a message M^*. The challenger chooses a random bit b, runs $\sigma^* \longleftarrow \mathsf{GSig}(gpk, gsk_{i_b}, M^*)$, and sends σ^* to $\mathcal{A}$. $\mathcal{A}$ is not allowed to request a challenge with a message M^* whose token is previously queried.

(iv) *Token Query (Phase II)*: after receiving the challenge $\mathcal{A}$ is again allowed to issue token queries. This time querying a token for M^* is disallowed.

(v) *Guess*: finally $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins the game if $b = b'$. $\mathcal{A}$ wins the game if $b = b'$.

The advantage of $\mathcal{A}$ is defined by $|\Pr[b = b'] - 1/2|$ and is denoted by $\mathrm{Adv}^{\text{opener-anony}}_{\mathcal{A}}(\lambda)$. We also say that a GS-MDO scheme has *opener anonymity with k-bounded tokens* if any probabilistic polynomial-time adversary $\mathcal{A}$ which issues at most k token queries in total has negligible advantage.

3.2.2. Admitter Anonymity. We next give the definition of anonymity against the admitter, which we call *admitter anonymity*. It is formalized in a similar manner to opener anonymity. That is, admitter anonymity requires signatures of two different signers to be indistinguishable even when the adversary is given the admitting key. We emphasize that our definition of opener anonymity is categorized as so-called *CCA-anonymity*. This means that the adversary in the security game is allowed to access the opening oracle. The formal definition is as follows:

Definition 2. A GS-MDO scheme is said to have *admitter anonymity* if for any probabilistic polynomial-time adversary $\mathcal{A}$ the advantage in the following game is negligible:

(i) *Setup*: the challenger runs $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ to obtain $(gpk, ok, ak, (gsk_i)_{i\in[n]})$. The challenger sends $(gpk, ak, (gsk_i)_{i\in[n]})$ to $\mathcal{A}$.

(ii) *Open Query (Phase I)*: $\mathcal{A}$ adaptively issues open queries (M, σ) in an arbitrary order. The challenger finds a recorded t_M which is a token for M, and if not found, runs $t_M \longleftarrow \mathsf{Td}(gpk, ak, M)$, and stores t_M. The challenger runs $i \longleftarrow \mathsf{GSig}(gpk, ok, M, \sigma, t_M)$ and returns i to $\mathcal{A}$.

(iii) *Challenge*: at some point $\mathcal{A}$ requests a challenge for i_0, $i_1 \in [n]$ and a message M^*. The challenger chooses a random bit b, runs $\sigma^* \longleftarrow \mathsf{GSig}(gpk, gsk_{i_b}, M^*)$ and sends σ^* to $\mathcal{A}$.

(iv) *Open Query (Phase II)*: after receiving the challenge $\mathcal{A}$ is again allowed to issue open queries. This time querying an opening of (M^*, σ^*) is disallowed.

(v) *Guess*: finally $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins the game if $b = b'$.

The advantage of $\mathcal{A}$ is defined by $|\Pr[b = b'] - 1/2|$ and is denoted by $\mathrm{Adv}^{\text{admitter-anony}}_{\mathcal{A}}(\lambda)$.

Notice that the number of opening queries the adversary issues is unbounded (but of course polynomially many).

3.2.3. Traceability. The last notion is *traceability*, which requires that even if the opener and the admitter collude and they further adaptively corrupt some group members, the corrupted parties can produce neither forged signatures nor untraceable signatures. We stress that in contrast to the case of the anonymity notions, this case considers the collusion of two authorities.

Definition 3. A GS-MDO scheme is said to have *traceability* if for any probabilistic polynomial-time adversary $\mathcal{A}$ the advantage in the following game is negligible:

(i) *Setup*: the challenger runs $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ to obtain $(gpk, ok, ak, (gsk_i)_{i\in[n]})$. The challenger sends (gpk, ok, ak) to $\mathcal{A}$.

(ii) *Query*: $\mathcal{A}$ adaptively issues the following two types of queries:

(1) $\mathcal{A}$ issues a *key revealing query* i. The challenger returns gsk_i to $\mathcal{A}$.

(2) $\mathcal{A}$ issues a *signing query* (i, M). The challenger runs $\sigma \longleftarrow \mathsf{GSig}(gpk, gsk_i, M)$ and returns σ.

(iii) *Forge*: finally $\mathcal{A}$ outputs a forgery (M^*, σ^*). $\mathcal{A}$ wins if $\mathsf{GVf}(gpk, M^*, \sigma^*) = \top$ and one of the following two conditions holds:

(a) $\mathsf{Open}(gpk, ok, M^*, \sigma^*, \mathsf{Td}(gpk, ak, M^*)) = \perp$, or

(b) all the following conditions hold,

* $\mathsf{Open}(gpk, ok, M^*, \sigma^*, \mathsf{Td}(gpk, ak, M^*)) = i^* \neq \perp$, and
* neither a key revealing query for the user i^* nor a signing query for (i^*, M^*) is submitted.

The advantage of $\mathcal{A}$ is defined by $\Pr[\mathcal{A}$ wins$]$ and denoted by $\mathrm{Adv}^{\text{trace}}_{\mathcal{A}}(\lambda)$.

4. Difficulty in Designing Efficient Constructions

In this section we discuss several difficulties in designing efficient GS-MDO schemes. We first observe the relationships between GS-MDO and other cryptographic primitives, and then we discuss the difficulty that lies in designing efficient constructions.

Regarding the relationship with other primitives, we show that the existence of a GS-MDO scheme implies that of an IBE scheme. In other words, we will present a black-box

$\mathsf{ISetup}(1^\lambda)$:
  $(gpk, ok, ak, (gsk_1, gsk_2)) \longleftarrow \mathsf{GKg}(1^\lambda, 1^2)$:
  $par \longleftarrow (gpk, ok, gsk_1, gsk_2); mk \longleftarrow ak$
  Output (par, mk).
$\mathsf{IEnc}(par, ID)$:
  For $i \in [\lambda]$:
    $K_i \longleftarrow \{0,1\}$
    $\sigma_i \longleftarrow \mathsf{GSig}(gpk, gsk_{K_i+1}, ID)$
  $C \longleftarrow (\sigma_1, \ldots, \sigma_\lambda)$
  $K \longleftarrow K_1 \| \cdots \| K_\lambda$
  Output (C, K).
$\mathsf{IExt}_{mk}(ID)$:
  $dk_{ID} \longleftarrow \mathsf{Td}(gpk, mk, ID)$
  Output dk_{ID}.
$\mathsf{IDec}(dk_{ID}, ID, C)$:
  Parse C as $(\sigma_1, \ldots, \sigma_\lambda)$
  For $i \in [\lambda]$:
    $j_i \longleftarrow \mathsf{Open}(gpk, ok, ID, \sigma_i, dk_{ID})$
    $K_i \longleftarrow j_i - 1$
  If $j_i \notin \{1, 2\}$ for some i
    then Output $\perp$
  Else Output $K_1 \| \cdots \| K_\lambda$.

Box 1: The black-box construction of an identity-based KEM from GS-MDO.

construction of IBE from any GS-MDO scheme. The same holds for the k-resilient versions.

This fact means that constructing a GS-MDO scheme is harder than IBE. Moreover, IBE is well known as a strong primitive. For example, there is no black-box construction of IBE from a trapdoor permutation or chosen-ciphertext secure PKE [55]. Therefore, GS-MDO is also strong and difficult to construct.

We note that Box 1 only shows a construction of an identity-based *key encapsulation mechanism* rather than identity-based *encryption*. However, it suffices since we can obtain a secure encryption scheme by combining the construction with an appropriate data encapsulation mechanism.

The formal theorems are as follows:

Theorem 4. *If the underlying GS-MDO scheme satisfies opener anonymity, the identity-based KEM in Box 1 is fully secure.*

Theorem 5. *If the underlying GS-MDO scheme satisfies opener anonymity with k-bounded tokens, the identity-based KEM in Box 1 is k-resilient.*

The intuition behind the construction in Box 1 is as follows. In the IEnc algorithm, each σ_i is a group signature of either gsk_0 or gsk_1. Therefore, by the opener anonymity of the GS-MDO scheme, the entity who does not have mk (corresponding to the admitter's key) cannot identify which signing key is used to make σ_i. The receiver can get dk_{ID} (corresponding to the message-dependent token), and only the receiver can open the group signature and get K_i by using the dk_{ID}. Formal proofs can be given by a straightforward modification of the proof by Abdalla and Warinschi [56] or the similar technique to that used by Ohtake, Fujii, Hanaoka, and Ogawa [57], hence we omit detailed proofs.

From Theorems 4 and 5, we make the following observation about the difficulty in constructing GS-MDO.

(i) *Inevitability of using IBE:* these theorems suggest that using IBE is essential for constructing a GS-MDO scheme. Considering the fact that a black-box construction of IBE from a trapdoor permutation is impossible [55], we can conclude that it is almost unavoidable for a GS-MDO scheme to rely on an IBE scheme or its equivalence, not only on a trapdoor permutation and an NIZK proof system. Otherwise one could construct an IBE scheme from surprisingly weaker primitives.

(ii) *Difficulty in constructing efficient GS-MDO in the standard model:* another important aspect of establishing an *efficient* GS-MDO scheme in the standard model is the necessity of realizing a "Groth-Sahai compatible" IBE scheme. This is because the only known construction of NIZK proof systems with reasonable efficiency in the standard model is limited to the Groth-Sahai proof system. Also note that an NIZK proof system has been an important building block for almost all group signature schemes ever.

We show a generic construction of GS-MDO using a NIZK proof system and IBE in Section 5. However, currently no known IBE scheme is Groth-Sahai compatible in the sense that the Groth-Sahai proof system cannot prove a kind of the well-formedness of an IBE ciphertext in zero-knowledge. (Libert and Joye [7] proposed an IBE scheme where we are able to prove the well-formedness of a ciphertext using the Groth-Sahai proof system. However, at the time of the publication of the conference version, there were no known IBE scheme with this property). This is because an IBE scheme in a bilinear group is typically able to encrypt a target-group element. In addition, the ciphertext of such a scheme includes a target-group element. Unfortunately, the Groth-Sahai proof system is not always able to prove a statement involving target-group elements *in zero-knowledge*.

Here we provide two methods of overcoming this gap. The first one is to adopt k-resilient IBE instead of fully secure IBE. In particular, we design a k-resilient IBE scheme from the DLIN assumption by modifying the Heng-Kurosawa scheme [8] for this purpose. (We also note that a similar construction can be obtained from the key-insulated encryption scheme proposed by Dodis, Katz, Xu, and Yung [58]). The construction of GS-MDO using k-resilient IBE is shown in Section 6.

The second method is to apply random oracles. We construct a GS-MDO scheme based on the BBS group signature scheme [10], which is one of the most efficient group signature schemes in the random oracle model. The opening procedure is implemented by the linear encryption scheme [10, 13], and a user's certificate is implemented by the Boneh-Boyen short signature scheme [27]. The MDO property is realized by adopting the BF IBE scheme [11]. In order to combine the short group signature scheme and the

BF IBE scheme, we replace the linear encryption scheme with a certain type of 2-out-of-2 multiple encryption. This construction is shown in Section 7.

5. Generic Construction

In this section, we give a generic construction of a GS-MDO scheme. The construction is built on an EUF-CMA secure signature scheme, a strongly unforgeable one-time signature scheme, a selective-tag weakly chosen-ciphertext secure tag-based KEM, a k-resilient identity-based KEM, and an adaptive NIZK proof system.

At a first glance, there are various building blocks. However, our generic construction only relies on the existence of an IBE scheme and that of an NIZK proof system. Indeed, signature schemes and a chosen-ciphertext secure tag-based encryption scheme can be constructed from a fully secure IBE scheme.

5.1. Underlying Ideas. The proposed construction shares an underlying idea with the generic construction of Bellare, Micciancio, and Warinschi (the BMW construction) [3] except that we do not need "simulation-soundness" for the underlying NIZK proof system. Instead of this strong security requirement, we combine (ordinary) NIZK proofs with a strongly unforgeable one-time signature scheme. We remark that essentially the same techniques have been used in a variety of contexts. To name a few, Groth [32] used this technique for an efficient group signature scheme in a very similar manner. Camenisch, Chandran, and Shoup [59] used it to construct simulation-sound NIZK proofs, improving the result of Groth [29].

5.2. Construction. In this section, we present our generic construction of GS-MDO. Our construction uses an identity-based KEM (ISetup, IExt, IEnc, IDec), tag-based KEM (TKg, TEnc, TDec), an EUF-CMA secure signature scheme (SigKg, Sign, Verify), a one-time signature scheme ($\mathsf{SigKg}^{\mathrm{OT}}$, $\mathsf{Sign}^{\mathrm{OT}}$, $\mathsf{Verify}^{\mathrm{OT}}$), and an NIZK proof system $(K_{\mathrm{NIZK}}, P_{\mathrm{NIZK}}, V_{\mathrm{NIZK}})$. In addition, we require that the session key spaces $\mathscr{K}_{\mathrm{PKE}}$ and $\mathscr{K}_{\mathrm{IBE}}$ be the same set, and be associated with a group operation $\odot$.

In our construction, a group member has a key pair (vk_i, sk_i) of the signature scheme, in which vk_i is authorized by another verification key vk_{issue} at the setup time. When a member generates a group signature, the member simply signs a message by sk_i. To be anonymous, the member further encrypts the signature together with the certificate (of the member), which authorizes the verification key vk_i, and attaches a noninteractive proof that demonstrates that a signature of an authorized member has been encrypted. To encrypt a signature, the member uses a multiple encryption technique to ensure that neither the opener nor the admitter can reveal the identity as long as the admitter has not issued a token to the opener. We need two requirements on the session key spaces of the tag-based KEM and the (k-resilient) IBE scheme. We require that $\mathscr{K}_{\mathrm{PKE}}$ be equal to $\mathscr{K}_{\mathrm{IBE}}$ as a set, and that they form a finite group with group operation $\odot$. These requirements are needed because we use a one-time pad to encrypt a signature of the group member. This group operation also needs to fall into the class of relations that can be represented by the used noninteractive proof system.

Let us explain the noninteractive proof that appears in the construction. The signature of the proposed scheme is of the form $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$, and, as mentioned above, the proof π demonstrates that a valid signature of an authorized group member has been encrypted into $(C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi)$ in a kind of a "multiple encryption" manner. More specifically, the proof π proves that there exist a randomness r (for tag-based KEM), another randomness ρ (for identity-based KEM), a user index i, and the verification key vk_i, a certificate $cert_i$, and a signature s on a message M, such that

$$\begin{aligned}
&(C_{\mathrm{PKE}}, K_{\mathrm{PKE}}) = \mathsf{TEnc}\left(pk, vk_{\mathrm{OT}}; r\right),\\
&(C_{\mathrm{IBE}}, K_{\mathrm{IBE}}) = \mathsf{IEnc}\left(par, M; \rho\right),\\
&\chi = \langle i, vk_i, cert_i, s\rangle \odot K_{\mathrm{PKE}} \odot K_{\mathrm{IBE}},\\
&\mathsf{Verify}\left(vk_{\mathrm{issue}}, \langle i, vk_i\rangle, cert_i\right) = \top,\\
&\mathsf{Verify}\left(vk_i, M, s\right) = \top.
\end{aligned} \tag{5}$$

Finally, two encoding functions are needed to complete the description of the generic construction. The first is used to encode the user index of a group member and his verification key into the message space of the signature scheme when generating certificates of group members. The second one is used to encode $(i, vk_i, cert_i, s)$ into $\mathscr{K}_{\mathrm{PKE}}$, where i is the user index of a group member and vk_i, $cert_i$, and s are his verification key, certificate, and signature, respectively. It is used when issuing a group signature, especially when encrypting his signature in order to hide his identity.

The complete description of the scheme is shown in Box 2 .

As stated in the following theorem, our generic construction has desirable security properties when all building blocks satisfy appropriate security properties.

Theorem 6. *The proposed scheme has opener anonymity with k-bounded tokens if the identity-based KEM is k-resilient and the noninteractive proof system is adaptively zero-knowledge. The proposed scheme satisfies opener anonymity if the identity-based KEM is fully secure and the noninteractive proof system is adaptively zero-knowledge.*

Theorem 7. *The proposed scheme has admitter anonymity when the a tag-based KEM is selective-tag weakly chosen-ciphertext secure, the noninteractive proof system is adaptively zero-knowledge, and the one-time signature scheme is strongly unforgeable.*

Theorem 8. *The proposed scheme has traceability when the noninteractive proof system is adaptively sound and the signature scheme is EUF-CMA secure.*

All the proofs of the theorems will appear in Appendix B.

$\mathsf{GKg}(1^\lambda, 1^n, 1^k)$:
  $(vk_{\text{issue}}, sk_{\text{issue}}) \longleftarrow \mathsf{SigKg}(1^\lambda)$
  $(pk, dk) \longleftarrow \mathsf{TKg}(1^\lambda)$
  $(par, mk) \longleftarrow \mathsf{ISetup}(1^\lambda, 1^k)$
  $\Sigma \longleftarrow K_{\text{NIZK}}(1^\lambda)$
  $gpk \longleftarrow (vk_{\text{issue}}, pk, par, \Sigma)$
  $ok \longleftarrow dk$
  $ak \longleftarrow mk$
  For all $i \in [n]$:
    $(vk_i, sk_i) \longleftarrow \mathsf{SigKg}(1^\lambda)$
    $cert_i \longleftarrow \mathsf{Sign}(sk_{\text{issue}}, \langle i, vk_i \rangle)$
    $gsk_i \longleftarrow (i, vk_i, cert_i, sk_i)$
  Output $(gpk, ok, ak, (gsk_i)_i)$.
$\mathsf{GSig}(gpk, gsk_i, M)$:
  Parse gpk as $(vk_{\text{issue}}, pk, par, \Sigma)$
  Parse gsk_i as $(i, vk_i, cert_i, sk_i)$
  $s \longleftarrow \mathsf{Sign}(sk_i, M)$
  $(vk_{\text{OT}}, sk_{\text{OT}}) \longleftarrow \mathsf{SigKg}^{\text{OT}}(1^\lambda)$
  $r \longleftarrow \{0,1\}^k$
  $\rho \longleftarrow \{0,1\}^k$
  $(C_{\text{PKE}}, K_{\text{PKE}}) \longleftarrow \mathsf{TEnc}(pk, vk_{\text{OT}})$
  $(C_{\text{IBE}}, K_{\text{IBE}}) \longleftarrow \mathsf{IEnc}(par, M)$
  $\chi \longleftarrow \langle i, vk_i, cert_i, s \rangle \odot K_{\text{PKE}} \odot K_{\text{IBE}}$
  $st \longleftarrow (vk_{\text{issue}}, pk, par, C_{\text{PKE}}, C_{\text{IBE}}, \chi)$
  $wt \longleftarrow (r, \rho, i, vk_i, cert_i, s)$
  $\pi \longleftarrow P_{\text{NIZK}}(\Sigma, st, wt)$
  $\sigma_{\text{OT}} \longleftarrow \mathsf{Sign}^{\text{OT}}(sk_{\text{OT}}, \langle C_{\text{PKE}}, C_{\text{IBE}}, \chi, \pi \rangle)$
  $\sigma \longleftarrow (vk_{\text{OT}}, C_{\text{PKE}}, C_{\text{IBE}}, \chi, \pi, \sigma_{\text{OT}})$
  Output σ.
$\mathsf{GVf}(gpk, M, \sigma)$:
  Parse gpk as $(vk_{\text{issue}}, pk, par, \Sigma)$
  Parse σ as $(vk_{\text{OT}}, C_{\text{PKE}}, C_{\text{IBE}}, \chi, \pi, \sigma_{\text{OT}})$
  $st \longleftarrow (vk_{\text{issue}}, pk, par, C_{\text{PKE}}, C_{\text{IBE}}, \chi)$
  If $\mathsf{Verify}^{\text{OT}}(vk_{\text{OT}}, \langle C_{\text{PKE}}, C_{\text{IBE}}, \chi, \pi \rangle, \sigma_{\text{OT}}) = \top$
  and $V_{\text{NIZK}}(\Sigma, st, \pi) = \top$
    then Output $\top$
  Else Output $\perp$.
$\mathsf{Td}(gpk, ak, M)$:
  Parse gpk as $(vk_{\text{issue}}, pk, par, \Sigma)$
  $t_M \longleftarrow \mathsf{IExt}(par, ak, M)$
  Output t_M.
$\mathsf{Open}(gpk, ok, M, \sigma, t_M)$:
  Parse gpk as $(vk_{\text{issue}}, pk, par, \Sigma)$
  Parse σ as $(vk_{\text{OT}}, C_{\text{PKE}}, C_{\text{IBE}}, \chi, \pi, \sigma_{\text{OT}})$
  $K_{\text{PKE}} \longleftarrow \mathsf{TDec}(ok, vk_{\text{OT}}, C_{\text{PKE}})$
  $K_{\text{IBE}} \longleftarrow \mathsf{IDec}(t_M, M, C_{\text{IBE}})$
  If $K_{\text{PKE}} = \perp$ or $K_{\text{IBE}} = \perp$
    then Output $\perp$
  $\langle i, vk_i, cert_i, s \rangle \longleftarrow \chi \odot K_{\text{IBE}}^{-1} \odot K_{\text{PKE}}^{-1}$
  $st \longleftarrow (vk_{\text{issue}}, pk, par, C_{\text{PKE}}, C_{\text{IBE}}, \chi)$
  If $\mathsf{Verify}^{\text{OT}}(vk_{\text{OT}}, \langle C_{\text{PKE}}, C_{\text{IBE}}, \chi, \pi \rangle, \sigma_{\text{OT}}) = \top$
  and $V_{\text{NIZK}}(\Sigma, st, \pi) = \top$ and $i \in [n]$
    then Output i
  Else Output $\perp$.

Box 2: The description of the proposed generic construction. In the concrete instantiation, $\langle \cdot \rangle$ denotes a tuple consisting of all group elements that appear in the bracket, and the operator $\odot$ is the component-wise group multiplication. The noninteractive proof system $(K_{\text{NIZK}}, P_{\text{NIZK}}, V_{\text{NIZK}})$ is for demonstrating the existence of a satisfying assignment of (5).

6. Construction in the Standard Model

In this section, we show an instantiation of GS-MDO in the standard model. Our scheme in this section satisfies the requirements of Section 3.2 under the DLIN assumption and the SFP assumption. Because of the incompatibility between the Groth-Sahai NIZK proof system and known (pairing-based) IBE schemes, we use k-resilient IBE and thus the MDO property is bounded to be k. Note that the instantiation is not straightforward even though we have the generic construction in Section 5, since it is not obvious whether there exist building blocks which fit the generic construction. Of course, if we use a general NIZK proof system, we obtain an instantiation, but it is far from an efficient construction.

6.1. Underlying Ideas. Towards an efficient scheme, we discuss how to instantiate the building blocks used in our generic construction in the previous section.

Regarding the noninteractive proof system, an obvious choice is the Groth-Sahai proof system, since there is no known practical construction of an NIZK proof system except the Groth-Sahai proof system. However, upon adopting the Groth-Sahai proof system, due to the limitation of the type of theorems that the Groth-Sahai proof system can prove, other building blocks are subjected to restrictions. More specifically, other building blocks need to be *structure preserving* [60], and in particular, a statement to be proven should not involve elements of $\mathbb{G}_T$, where $\mathbb{G}_T$ is the target group of the underlying bilinear groups. Hence, we have to choose an IBE scheme which fulfills this requirement as a building block, but unfortunately, no such scheme is known (except for the Libert-Joye scheme). This means that *it is not straightforward to construct an efficient instantiation of our generic construction from the Groth-Sahai proof system*.

In this section, we give an efficient instantiation by constructing a *structure-preserving* IBE scheme and choosing other appropriate building blocks. However, we must also honestly mention that our IBE does not provide full security but only k-resilience [8].

Our structure-preserving k-resilient IBE scheme is obtained by means of modifying the Heng-Kurosawa scheme [8] which is secure in the sense of k-resilient security under the decision Diffie-Hellman (DDH) assumption. Since the DDH assumption does not hold in (symmetric) bilinear groups, it is not possible to utilize it as it is, and thus, we construct a modified version of this scheme which is secure under the DLIN assumption.

6.2. Construction. As described above, an efficient instantiation is not straightforward even from our generic construction in Section 5, since it is not obvious whether there exist schemes which can be plugged into the generic construction. Therefore, we select concrete building blocks for the instantiation:

(i) *Groth-Sahai Proofs [5, 61].* This is an efficient noninteractive proof system for groups with a bilinear map. This proof system can demonstrate the validity of a broad range of algebraic equations in a

zero-knowledge manner and is useful for avoiding an expensive blowup from general NIZK techniques. We choose the DLIN-based instantiation of the Groth-Sahai proof system, which was proved to be perfectly sound with adaptive zero-knowledge from the DLIN assumption.

(ii) *Abe-Haralambiev-Ohkubo Signature [6, 60]*. This is a structure-preserving signature scheme, in the sense that the signing and verification procedures have no use of nonalgebraic operations. This property is essential when the scheme is used together with the Groth-Sahai proof system that has the restriction that it is unable to treat nonalgebraic relations such as a verification equation of a signature scheme which involves hashing. This signature scheme was proven secure from the SFP assumption.

(iii) *DLIN Variant of Cramer-Shoup scheme [13]*. The Groth-Sahai proof system is highly reliant on its use of pairing operations, and thus we need to choose the type of pairing on which we base our scheme. Type III or II may allow an efficient instantiation. However, in this setting a signature of the Abe-Haralambiev-Ohkubo scheme is bilateral, namely, it contains elements of both source groups. To encrypt such a bilateral message, we need to set up the encryption scheme in both source groups, which makes the scheme inefficient. Hence, we choose all the building blocks to be instantiated in Type I curves. For this reason, we choose a modification of Shacham's DLIN variant [13] of the Cramer-Shoup encryption scheme as the tag-based KEM. We modify this scheme to be tag-based KEM to fit into our generic construction. (A possible alternative choice here is Kiltz' tag-based encryption [52], which could reduce the size of NIZK proofs owning to its public verifiability. One drawback of this scheme is that, to the best of our knowledge, it does not allow the encryption of multiple group elements with a constant ciphertext overhead. On the other hand, the Cramer-Shoup scheme (and its DLIN variant by Shacham) allow such a modification. See Section C.3 for the details of this modification).

We then show a more concrete description of our generic construction when being instantiated with the above building blocks. Notice that a signature of our concrete instantiation based on the above building blocks, is of the following form:

$$\sigma = \left(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}}\right). \tag{6}$$

As explained above, χ is represented by 29 group elements, which we denote by $\chi = (\chi_1, \ldots, \chi_{29})$. A more detailed description of each element is as follows:

$$vk_{\mathrm{OT}} = \left(g^{s_0}, g^{s_1}, g^{x}\right)$$

$$C_{\mathrm{PKE}} = \left(u^{r}, v^{\widetilde{r}}, h^{r}h^{\widetilde{r}}, \left(XY^{t}\right)^{r}\left(\widetilde{X}\widetilde{Y}^{t}\right)^{\widetilde{r}}\right)$$

$$C_{\mathrm{IBE}} = \left(u^{\rho}, v^{\widetilde{\rho}}, h^{\rho}h^{\widetilde{\rho}}\right)$$

$$\chi_1 = Z_1^{r} \cdot \widetilde{Z}_1^{\widetilde{r}} \cdot \left(\prod_{j=0}^{k} D_{1,j}^{M^j}\right)^{\rho} \cdot \left(\prod_{j=0}^{k} \widetilde{D}_{1,j}^{M^j}\right)^{\widetilde{\rho}} \cdot m_1$$

$$\vdots$$

$$\chi_{29} = Z_{29}^{r} \cdot \widetilde{Z}_{29}^{\widetilde{r}} \cdot \left(\prod_{j=0}^{k} D_{29,j}^{M^j}\right)^{\rho} \cdot \left(\prod_{j=0}^{k} \widetilde{D}_{29,j}^{M^j}\right)^{\widetilde{\rho}} \cdot m_{29}$$

$$\sigma_{\mathrm{OT}} = \left(x + e \cdot s_0 + \left(\mathrm{CR}\left(\left\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi_1, \ldots, \chi_{29}, \pi\right\rangle\right) + e\right) \cdot s_1\right) \tag{7}$$

where CR is a collision resistant hash function and π is a Groth-Sahai proof.

In (7), $(m_1, \ldots, m_{29})$ is the encoding of $(i, vk_i, cert_i, s)$, $Z_i^{r} \cdot \widetilde{Z}_i^{\widetilde{r}}$ corresponds to the session key K_{PKE} of the tag-based KEM based on a DLIN variant of the Cramer-Shoup PKE scheme, and $(\prod_{j=0}^{k} D_{i,j}^{M^j})^{\rho} \cdot (\prod_{j=0}^{k} \widetilde{D}_{i,j}^{M^j})^{\widetilde{\rho}}$ corresponds to the session key K_{IBE} of the k-resilient identity-based KEM based on the Heng-Kurosawa IBE scheme. C_{PKE} and C_{IBE} are ciphertexts of the tag-based KEM and the identity-based KEM, respectively. σ_{OT} is Wee's one-time signature on $(C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi_1, \ldots, \chi_{29}, \pi)$.

The signature size of the construction in Section 6 is described in Table 1. The column "Cost" is the number of group elements of a statement to be proven, and the number of group elements of the Groth-Sahai NIZK proof per a statement. The abbreviations "var(s).", "lin. eq.", "mult", and "pair. prod." mean "variable(s)", "linear equation", "multiplication" and "pairing product", respectively. The total number of group elements is shown in the column "$\mathbb{G}$". As shown in Table 1, the standard model scheme achieves reasonably efficient performance.

From Theorems 6, 7, and 8, the following theorems hold.

Theorem 9. *When instantiating the identity-based KEM and the NIZK proof system in Box 2 with our DLIN variant of the Heng-Kurosawa k-resilient IBE scheme and the Groth-Sahai proof system, the resulting scheme satisfies opener anonymity with k-bounded tokens under the DLIN assumption.*

Theorem 10. *When instantiating the tag-based KEM, the NIZK proof system and the one-time signature scheme in Box 2 with the DLIN variant of the Cramer-Shoup PKE, the Groth-Sahai proof system and the Wee one-time signature scheme, the resulting scheme satisfies admitter anonymity under the DLIN assumption.*

Theorem 11. *When instantiating the signature scheme in Box 2 with the Abe-Haralambiev-Ohkubo signature scheme, the resulting scheme satisfies traceability under the SFP assumption.*

Table 1: The number of group elements of GS-MDO in the standard model.

	Cost	$\mathbb{G}$	$\mathbb{Z}_p$
verification key vk_{sots}		3	
TB-KEM CT C_{PKE}		4	
IB-KEM CT C_{IBE}		3	
DEM CT χ		29	
NIZK proof			
commitments for			
$r, \widetilde{r} \in \mathbb{Z}_p$	2 vars. × 3|$\mathbb{G}$| per var.	6	
$\rho, \widetilde{\rho} \in \mathbb{Z}_p$	2 vars. × 3|$\mathbb{G}$| per var.	6	
$g^i \in \mathbb{G}$	1 var. × 3|$\mathbb{G}$| per var.	3	
$vk_i \in \mathbb{G}^{14}$	14 vars. × 3|$\mathbb{G}$| per var.	42	
$cert_i \in \mathbb{G}^7$	7 vars. × 3|$\mathbb{G}$| per var.	21	
$s \in \mathbb{G}^7$	7 vars. × 3|$\mathbb{G}$| per var.	21	
equations for			
well-formedness of C_{PKE}	4 $\mathbb{Z}_p$-var. lin. eqs. × 2|$\mathbb{G}$| per eq.	8	
well-formedness of C_{IBE}	3 $\mathbb{Z}_p$-var. lin. eqs. ×× 2|$\mathbb{G}$| per eq.	6	
well-formedness of χ	29 multi-scalar mult. eqs. × 9|$\mathbb{G}$| per eq.	261	
validity of $cert_i$	2 pair. prod. eqs. × 9|$\mathbb{G}$| per eq.	18	
validity of s	2 pair. prod. eqs. × 9|$\mathbb{G}$| per eq.	18	
one-time signature σ_{OT}			2
Total		449	2

7. Construction in the Random Oracle Model

The previous section explained a GS-MDO scheme in the standard model, but it only allows at most k tokens to be issued. In this section, we remove this restriction by using random oracles, and present a concrete GS-MDO scheme that allows an unbounded number of tokens. Moreover, its signature size is 1/20 times smaller than that of the scheme in Section 6, and it is also secure under the DLIN, DBDH, and q-SDH assumptions. We give the description of the proposed scheme in Box 3.

7.1. Underlying Ideas. Our standard model instantiation uses the Groth-Sahai proof system as the NIZK proof system, and therefore the building block identity-based KEM needs to be compatible with the Groth-Sahai proof system (see Section 6.1). Unfortunately, the only IBE scheme with this property is Libert et al.'s [7], however, this scheme has a ciphertext containing a linear number of group elements in the security parameter. We overcome this weakness by avoiding the use of the Groth-Sahai proof system. More specifically, we construct an NIZK proof system in the random oracle model based on the Fiat-Shamir heuristics, and choose the building block IBE scheme that is fully secure (i.e., fully collusion resilient).

We start with the Boneh-Boyen-Shacham (BBS) group signature scheme [10], which is one of the most popular (standard) group signature schemes, to achieve the above idea. Each group member in this scheme is provided with an (ordinary) signature [27] of the Boneh-Boyen signature scheme, which certifies that the owner is a group member. A group signature of this scheme consists of two parts. The first part involves the linear encryption scheme of the certificate, whereas the second part involves the "signature of knowledge" [62] of the encrypted certificate. The decryption key for the linear encryption scheme is held by the opener, with which he can revoke the anonymity of any group signature.

We extend the BBS group signature scheme by replacing the linear encryption scheme with a certain type of 2-out-of-2 multiple encryption of ordinary PKE and IBE schemes. The multiple encryption is designed to ensure that one can decrypt an entire ciphertext only when he is able to decrypt ciphertexts of *both* the PKE scheme and the IBE scheme. Such multiple encryption can be accomplished by simple 2-out-of-2 secret sharing.

This feature enables us to achieve the MDO property as follows. In our construction, the opener only possesses a decryption key for the PKE scheme, and the admitter holds the master secret of the IBE scheme. To anonymously sign a message M, a signer encrypts his certificate using M as the identity of the IBE encryption. A decryption key (of the IBE scheme) under a message M can serve as a message-specific token for M.

7.2. Construction. In this subsection, we describe the proposed GS-MDO scheme in the random oracle model in detail. The complete description of the proposed scheme is shown in Box 3. As described above, our scheme is constructed by modifying the BBS group signature scheme in such a way that the linear encryption scheme in the BBS group signature scheme is replaced with 2-out-of-2 multiple encryption by PKE and IBE, which are instantiated with the linear encryption scheme and the BF IBE scheme

$\mathsf{GKg}\ (1^\lambda, 1^n)$:
$(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathcal{G}(1^\lambda)$
$u, v, h \longleftarrow \mathbb{G} \setminus \{1\}$
$\xi_1, \xi_2, \xi_3, \zeta, \gamma \longleftarrow \mathbb{Z}_p$
$g_1 \longleftarrow u^{\xi_1} h^{\xi_3}$; $g_2 \longleftarrow v^{\xi_2} h^{\xi_3}$; $y \longleftarrow g^\zeta$; $w \longleftarrow g^\gamma$
For $i \in [n]$:
$x_i \longleftarrow \mathbb{Z}_p$
$A_i \longleftarrow g^{1/(\gamma + x_i)}$
$gpk \longleftarrow (p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w, H_1, H_2)$
$ok \longleftarrow (\xi_1, \xi_2, \xi_3, (e(A_i, g))_{i \in [n]})$
$ak \longleftarrow \zeta$
For $i \in [n]$:
$gsk_i \longleftarrow (A_i, x_i)$
Output $(gpk, ok, ak, (gsk_i)_{i \in [n]})$.
$\mathsf{GSig}\ (gpk, i, gsk_i, M)$:
$(p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w, H_1, H_2) \longleftarrow gpk$
$(A_i, x_i) \longleftarrow gsk_i$
$\alpha, \beta, \rho, \eta \longleftarrow \mathbb{Z}_p$
$(T_1, T_2, T_3, T_4) \longleftarrow (u^\alpha, v^\beta, h^{\alpha+\beta}, g_1^\alpha g_2^\beta A_i g^\eta)$
$(T_5, T_6) \longleftarrow (g^\rho, e(y, H_1(M))^\rho e(g, g)^{-\eta})$
$r_\alpha, r_\beta, r_\rho, r_\eta, r_x, r_{\alpha x}, r_{\beta x}, r_{\rho x}, r_{\eta x} \longleftarrow \mathbb{Z}_p$
$R_1 \longleftarrow u^{r_\alpha}$
$R_2 \longleftarrow v^{r_\beta}$
$R_3 \longleftarrow h^{r_\alpha + r_\beta}$
$R_4 \longleftarrow e(T_4, g)^{r_x} e(g_1, w)^{-r_\alpha} e(g_1, g)^{-r_{\alpha x}}$
$\cdot e(g_2, w)^{-r_\beta} e(g_2, g)^{-r_{\beta x}}$
$\cdot e(g, w)^{-r_\eta} e(g, g)^{-r_{\eta x}}$
$R_5 \longleftarrow g^{r_\rho}$
$R_6 \longleftarrow e(y, H_1(M))^{r_\rho} e(g, g)^{-r_\eta}$
$R_7 \longleftarrow T_1^{r_x} u^{-r_{\alpha x}}$
$R_8 \longleftarrow T_2^{r_x} v^{-r_{\beta x}}$
$R_9 \longleftarrow T_5^{r_x} g^{-r_{\rho x}}$
$R_{10} \longleftarrow T_6^{r_x} e(y, H_1(M))^{-r_{\rho x}} e(g, g)^{r_{\eta x}}$
$c \longleftarrow H_2(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$
$s_\alpha \longleftarrow r_\alpha + c\alpha$; $s_\beta \longleftarrow r_\beta + c\beta$
$s_\rho \longleftarrow r_\rho + c\rho$; $s_\eta \longleftarrow r_\eta + c\eta$
$s_x \longleftarrow r_x + cx_i$; $s_{\alpha x} \longleftarrow r_{\alpha x} + c\alpha x_i$
$s_{\beta x} \longleftarrow r_{\beta x} + c\beta x_i$; $s_{\rho x} \longleftarrow r_{\rho x} + c\rho x_i$
$s_{\eta x} \longleftarrow r_{\eta x} + c\eta x_i$
$\sigma \longleftarrow (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$
Output σ.
$\mathsf{GVf}\ (gpk, M, \sigma)$:
$(p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w, H_1, H_2) \longleftarrow gpk$
$(T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x}) \longleftarrow \sigma$
$R_1' \longleftarrow u^{s_\alpha} T_1^{-c}$
$R_2' \longleftarrow v^{s_\beta} T_2^{-c}$
$R_3' \longleftarrow h^{s_\alpha + s_\beta} T_3^{-c}$
$R_4' \longleftarrow e(T_4, g)^{s_x} e(g_1, w)^{-s_\alpha} e(g_1, g)^{-s_{\alpha x}}$
$\cdot e(g_2, w)^{-s_\beta} e(g_2, g)^{-s_{\beta x}}$
$\cdot e(g, w)^{-s_\eta} e(g, g)^{-s_{\eta x}}$
$\cdot (e(g, g)/e(T_4, w))^{-c}$
$R_5' \longleftarrow g^{s_\rho} T_5^{-c}$
$R_6' \longleftarrow e(y, H_1(M))^{s_\rho} e(g, g)^{-s_\eta} T_6^{-c}$
$R_7' \longleftarrow T_1^{s_x} u^{-s_{\alpha x}}$
$R_8' \longleftarrow T_2^{s_x} v^{-s_{\beta x}}$
$R_9' \longleftarrow T_5^{s_x} g^{-s_{\rho x}}$
$R_{10}' \longleftarrow T_6^{s_x} e(y, H_1(M))^{-s_{\rho x}} e(g, g)^{s_{\eta x}}$
If $c = H_2(M, T_1, \ldots, T_6, R_1', \ldots, R_{10}')$
Output $\top$
Else output $\bot$.

Box 3: Continued.

$\mathsf{Td}\ (gpk, ak, M)$:
$(p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w, H_1, H_2) \longleftarrow gpk$
$\zeta \longleftarrow ak$
Output $t_M \longleftarrow H_1(M)^\zeta$.
$\mathsf{Open}\ (gpk, ok, M, \sigma, t_M)$:
$(p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w, H_1, H_2) \longleftarrow gpk$
$(\xi_1, \xi_2, \xi_3, (e(A_i, g))_{i \in [n]}) \longleftarrow ok$
$(T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x}) \longleftarrow \sigma$
If $\mathsf{GVf}(gpk, M, \sigma) = \bot$
Output $\bot$
If $\exists i \in [n]$: $e(T_4/T_1^{\xi_1} T_2^{\xi_2} T_3^{\xi_3}, g) \cdot T_6/e(T_5, t_M) = e(A_i, g)$
Output i;
Else output $\bot$.

Box 3: The proposed construction in the random oracle model. The functions H_1 and H_2 are cryptographic hash functions $H_1 : \{0,1\}^* \longrightarrow \mathbb{G}$ and $H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_p$, which are modeled as random oracles in the security proof. $\mathcal{G}$ is the parameter generation algorithm, which is defined in Section 2.5.

[11], respectively. Specifically, the values $(T_1, T_2, T_3, T_4, T_5, T_6)$ generated in the signing algorithm are of the following form

$$\begin{aligned} T_1 &= u^\alpha, \\ T_2 &= v^\beta, \\ T_3 &= h^{\alpha+\beta}, \\ T_4 &= {g_1}^\alpha {g_2}^\beta A g^\eta, \\ T_5 &= g^\rho, \\ T_6 &= e\left(y, H_1(M)\right)^\rho e\left(g, g\right)^{-\eta}, \end{aligned} \tag{8}$$

where the first four components constitute (a chosen-ciphertext secure variant of) the linear encryption of a 2-out-of-2 secret sharing of A. Here α and β are the randomness for the linear encryption scheme and η is the randomness for the 2-out-of-2 secret sharing. The other components, T_5 and T_6, constitute an encryption by the BF IBE scheme of the other share of the 2-out-of-2 secret sharing.

As mentioned above, our linear encryption scheme is not identical to that used in the BBS group signature scheme and is modified to achieve CCA-anonymity. It is obtained by modifying the linear encryption scheme (in the BBS scheme) to a linear analogue of the "double encryption" [20, 63, 64] scheme and proving the "well-formedness" by a Schnorr-type proof. This can be seen as a variant of (the DLIN variant [13] of) the Cramer-Shoup encryption scheme [12, 65] in which the validity-check property is realized by a Schnorr-type proof.

Note that the BF IBE scheme is also slightly modified in such a way that the scheme does not use a hash function for deriving the DEM key that masks a plaintext. Instead of using a hash function, we employ the slightly stronger assumption of the DBDH assumption (rather than the computational variant, on which the original BF scheme is based). This

is due to the incompatibility of the use of a hash function of this type and Schnorr-type proofs. We also remark that our anonymity notion does not require the IBE scheme to be chosen-ciphertext secure, and hence we use a chosen-plaintext secure version of the BF scheme.

We further remark that $R_1, \ldots, R_{10}, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}$, and $s_{\eta x}$ in Box 3 come from a Schnorr-type protocol that proves the knowledge of α, β, ρ, η, and x satisfying the equations

$$\begin{aligned} T_1 &= u^\alpha, \\ T_2 &= v^\beta, \\ T_3 &= h^{\alpha+\beta}, \\ e(g,g) &= e\left(T_4 {g_1}^{-\alpha} {g_2}^{-\beta} g^{-\eta}, w g^x\right), \\ T_5 &= g^\rho, \\ T_6 &= e(y, H_1(M))^\rho e(g,g)^{-\eta}. \end{aligned} \tag{9}$$

The first three equations prove the knowledge of α and β that are used for the linear encryption part, and furthermore prove the "well-formedness" of the ciphertext (in other words, they prove that (T_1, T_2, T_3) constitutes a linear tuple). We note that the latter (proving the "well-formedness") is crucial for achieving CCA-anonymity. The last two equations prove the knowledge of ρ and η that are used for the BF IBE scheme. Here ρ is the randomness for the BF IBE scheme, and η is the randomness for the 2-out-of-2 secret sharing. The fourth equation proves that the group element encrypted in T_4 satisfies the verification equation of the Boneh-Boyen signature scheme. Recall that $T_4 = g_1^\alpha g_2^\beta A_i g^\eta$, where A_i is a Boneh-Boyen signature which certifies the membership of the signer. In other words, this equation proves that when the randomness that hides A_i is removed from T_4, T_4 is verified as valid by the verification equation of the Boneh-Boyen signature scheme.

More concretely, introducing four intermediate variables $\delta_1, \delta_2, \delta_3$, and δ_4 (which are intended to be $\delta_1 = \alpha x$, $\delta_2 = \beta x$, $\delta_3 = \rho x$, and $\delta_4 = \eta x$), the underlying protocol proves the knowledge of α, β, ρ, η, x, δ_1, δ_2, δ_3, and δ_4 satisfying the equations

$$T_1 = u^\alpha, \tag{10a}$$

$$T_2 = v^\beta, \tag{10b}$$

$$T_3 = h^{\alpha+\beta}, \tag{10c}$$

$$\begin{aligned} \frac{e(g,g)}{e(T_4,w)} &= e(T_4,g)^x e(g_1,w)^{-\alpha} e(g_1,g)^{-\delta_1} \\ &\cdot e(g_2,w)^{-\beta} e(g_2,g)^{-\delta_2} e(g,w)^{-\eta} e(g,g)^{-\delta_4}, \end{aligned} \tag{10d}$$

$$T_5 = g^\rho, \tag{10e}$$

$$T_6 = e(y, H_1(M))^\rho e(g,g)^{-\eta}, \tag{10f}$$

$$1 = T_1^x u^{-\delta_1}, \tag{10g}$$

$$1 = T_2^x v^{-\delta_2}, \tag{10h}$$

$$1 = T_5^x g^{-\delta_3}, \tag{10i}$$

$$1 = T_6^x e(y, H_1(M))^{-\delta_3} e(g,g)^{\delta_4}. \tag{10j}$$

We explain these ten equations one by one. The first three equations are exactly the same as those of (9). Equation (10d) comes from expanding the equation

$$e(g,g) = e\left(T_4 {g_1}^{-\alpha} {g_2}^{-\beta} g^{-\eta}, w g^x\right) \tag{11}$$

to obtain

$$\begin{aligned} e(g,g) &= e(T_4,w) e(T_4,g^x) e({g_1}^{-\alpha},w) e({g_1}^{\alpha},g^x) \\ &\cdot e\left({g_2}^{-\beta},w\right) e\left({g_2}^{-\beta},g^x\right) e(g^{-\eta},w) e(g^{-\eta},g^x), \\ &= e(T_4,w) e(T_4,g)^x e(g_1,w)^{-\alpha} e(g_1,g)^{-\alpha x} \\ &\cdot e(g_2,w)^{-\beta} e(g_2,g)^{-\beta x} e(g,w)^{-\eta} e(g,g)^{-\eta x}, \end{aligned} \tag{12}$$

replacing αx with δ_1, βx with δ_2, and ηx with δ_4 to obtain

$$\begin{aligned} e(g,g) &= e(T_4,w) e(T_4,g)^x e(g_1,w)^{-\alpha} e(g_1,g)^{-\delta_1} \\ &\cdot e(g_2,w)^{-\beta} e(g_2,g)^{-\delta_2} e(g,w)^{-\eta} \\ &\cdot e(g,g)^{-\delta_4}, \end{aligned} \tag{13}$$

and rearranging them to obtain

$$\begin{aligned} \frac{e(g,g)}{e(T_4,w)} &= e(T_4,g)^x e(g_1,w)^{-\alpha} e(g_1,g)^{-\delta_1} \\ &\cdot e(g_2,w)^{-\beta} e(g_2,g)^{-\delta_2} e(g,w)^{-\eta} \\ &\cdot e(g,g)^{-\delta_4}, \end{aligned} \tag{14}$$

which is identical to (10d). We then need to prove that the intermediate variables surely have appropriate values, namely, that $\delta_1 = \alpha x$, $\delta_2 = \beta x$, and $\delta_4 = \eta x$. Equations (10g)–(10j) are for this purpose. To prove $\delta_1 = \alpha x$, we introduce (10g). This equation, together with (10a), proves that $\delta_1 = \alpha x$. Specifically, (10a) ensures that T_1 is equal to u^α, and then (10g) in turn ensures that

$$1 = (u^\alpha)^x u^{-\delta_1}. \tag{15}$$

Since u is a generator, we have that

$$\alpha x - \delta_1 = 0, \tag{16}$$

which is what we want to prove. Similarly, (10h) together with (10b) proves that $\beta x = \delta_2$. To prove $\eta x = \delta_4$, we need to introduce another intermediate variable δ_3 which is intended to be equal to ρx. Towards this end, (10e) and (10i) prove that $\rho x = \delta_3$. Using this, (10f) and (10j) prove that $\delta_4 = \eta x$.

Table 2: Performance comparison among pairing-based GS-MDO and GS schemes.

	Signature size				
	# of $[\mathbb{G},\mathbb{Z}_p,\mathbb{G}_T]$-elements	bits	MDO	Assumptions	Without RO
Ours (Section 6)	[449, 2, 0]	79364	k-bounded	DLIN, SFP	Yes
Ours (Section 7)	[5, 10, 1]	3636	unbounded	DLIN, DBDH, q-SDH	No
Libert-Joye [7]	$[53\log n + 33, 2, 0]$	$9328\log N + 6148$	unbounded	DLIN, D3DH	Yes
BBS [10]	[3, 6, 0]	1548	-	DLIN, q-SDH	No
Groth [32]	[50, 0, 0]	8800	-	DLIN, q-SDH, q-U	Yes

Signature Size. The signature size in terms of the number of group elements and the bit sizes for 80-bit security. We adopt the estimates of the bit sizes of group elements used in [9] (the sizes of the elements of $\mathbb{G}$, $\mathbb{Z}_p$, and $\mathbb{G}_T$ are 176 bits, 170 bits, and 1056 bits, respectively.) n is the number of group members. We assume that the Wee one-time signature scheme [48] is used for the instantiation of the one-time signature scheme.
MDO. The security level of the message-dependent functionality, where "k-bounded" denotes that one needs, at the setup, to fix the number of tokens that will be issued, and "unbounded" denotes that one needs not fix the upper bound.
Assumptions. The hardness assumption on which the scheme is based. D3DH stands for the decision 3-party Diffie-Hellman assumption [11]. The q-U assumption is defined in [32].
Without RO. Whether or not the scheme requires the random oracle model.

Specifically, (10f) ensures that $T_6 = e(y, H_1(M))^{\rho} e(g,g)^{-\eta}$, and then (10j) in turn ensures that

$$1 = \left(e\left(y, H_1\left(M\right)\right)^{\rho} e\left(g,g\right)^{-\eta}\right)^{x} e\left(y, H_1\left(M\right)\right)^{-\delta_3} \cdot e\left(g,g\right)^{\delta_4}. \tag{17}$$

Since (10e) and (10i) ensure that $\delta_3 = \rho x$, canceling out $e(y, H_1(M))^{\rho x}$ and $e(y, H_1(M))^{-\delta_3}$, (10j) further ensures that

$$1 = e\left(g,g\right)^{-\eta x} e\left(g,g\right)^{\delta_4}, \tag{18}$$

since $e(g,g)$ is a generator, this equation ensures that

$$-\eta x + \delta_4 = 0, \tag{19}$$

which is what we want to prove.

The security of our proposed scheme is proved as follows.

Theorem 12. *If the DBDH assumption holds, the proposed construction has opener anonymity in the random oracle model.*

Theorem 13. *If the DLIN assumption holds, the proposed construction has admitter anonymity in the random oracle model.*

Theorem 14. *If the q-SDH assumption holds, the proposed construction has traceability in the random oracle model.*

The proofs of Theorems 12, 13, and 14 are given in Appendix D.

8. Efficiency Comparison with Pairing-Based Group Signatures

Finally, we give a brief efficiency comparison between the proposed schemes and existing pairing-based group signature schemes (with and without the MDO property). Table 2 shows a detailed comparison among such schemes.

The signature size of our standard model scheme is 10 times larger than that of Groth scheme [32]. We believe that this is a reasonably practical performance and can be implemented in practice.

Compared with our scheme in the standard model, our scheme in the random oracle model is improved in two aspects: the first is the removal of the a priori upper bound on the number of tokens the admitter can issue, and the second is the substantial reduction of the signature size. When a 170-bit prime-order group is used, the signature size of our GS-MDO scheme in the standard model is 79364 bits, and that of our GS-MDO scheme in the random oracle model is 3636 bits, while that of the BBS scheme [10] is 1548 bits. (We adopt the estimates of bit sizes for group elements used in [9]). We also include more details of the performance of these schemes in Table 2. These two improvements are achieved at the cost of using random oracles, as indicated in the "Without RO" column in the table.

In the Libert-Joye scheme, a signature contains $O(\log N)$ group elements, whereas our schemes contain a constant number of group elements. However, our standard model scheme does not achieve the unbounded MDO property. Constructing an unbounded GS-MDO scheme secure in the standard model with a constant-signature size is an interesting open problem.

Compared with the BBS scheme, we believe that the proposed scheme in the random oracle model achieves the MDO property at a reasonable cost given the increase in the signature size. As shown in Table 2, the signature size of the proposed scheme is almost twice that of the BBS scheme. We also note that our random oracle model scheme achieves CCA-anonymity, which guarantees a markedly higher level of security than CPA-anonymity, which is achieved by the BBS scheme.

9. Conclusion and Open Problems

In this paper we proposed a new anonymous authentication primitive called group signatures with message-dependent opening. This primitive is an extension of ordinary group signatures, which relaxes the strong assumption that the opener, who is able to trace the identity of the signer of a

signature, does not misuse his strong capability of breaking anonymity. The primitive does this by introducing another authority called the admitter, who is able to admit the opening of signatures on some specific messages to the opener.

We formalized this new primitive by providing the syntax and security requirements of the primitive. Furthermore, we discussed the difficulties in constructing an efficient instantiation of the primitive, in particular, showed that IBE is inevitable for an instantiation. We provided one generic construction and two specific constructions. The first specific construction was proven to be secure in the standard model, while the other was proven to be secure in the random oracle model. Finally, we compared two specific constructions and known group signature schemes (with and without the message-dependent opening property).

We list some open problems. We assumed that the setup of the scheme is executed honestly and that the entity who runs the setup does not collude with an adversary. An open problem is to provide a formal security requirement that does not require this assumption and an instantiation for it. Our schemes do not provide security against an adversary which may maliciously generate the opener's or the admitter's keys. Achieving this security is another open problem.

Appendix

A. Bellare and Neven's Forking Lemma

Here we recall Bellare and Neven's forking lemma [66].

Lemma A.1. *Fix an integer $q \geq 1$ and a set H of size $h \geq 2$. Let $\mathcal{I}$ be a randomized algorithm that we call the input generator. Let $\mathcal{A}$ be a randomized algorithm that takes x generated by $\mathcal{I}$ and $c_1,\ldots, c_q \in H$ as inputs, and returns a pair $(j,\sigma) \in \{0,\ldots,q\}\times\{0,1\}^*$. The accepting probability of $\mathcal{A}$, denoted* acc, *is defined as the probability that $j \geq 1$ in the experiment*

$$\begin{aligned} x &\longleftarrow \mathcal{I};\\ c_1,\ldots,c_q &\longleftarrow H;\\ (j,\sigma) &\longleftarrow \mathcal{A}\left(x,c_1,\ldots,c_q\right). \end{aligned} \tag{A.1}$$

The forking algorithm $\mathcal{F}_{\mathcal{A}}$ *associated to $\mathcal{A}$ is the randomized algorithm that takes input x and proceeds as follows:*

Algorithm $\mathcal{F}_{\mathcal{A}}(x)$:

Pick coins rnd for $\mathcal{A}$ at random

$c_1^*,\ldots,c_q^* \longleftarrow H$

$(j^*,\sigma^*) \longleftarrow \mathcal{A}\left(x,c_1^*,\ldots,c_q^*;\mathsf{rnd}\right)$

If $j^* = 0$ *then return* $(0,\perp,\perp)$

$c_{j^*}^{**},\ldots,c_q^{**} \longleftarrow H$

$(j^{**},\sigma^{**}) \longleftarrow \mathcal{A}\left(x,c_1^*,\ldots,c_{j^*-1}^*,c_{j^*}^{**},\ldots,c_q^{**};\mathsf{rnd}\right)$

If $\left(j^* = j^{**} \text{ and } c_{j^*}^* \neq c_{j^*}^{**}\right)$ *then*

return $(1,\sigma^*,\sigma^{**})$

Else

return $(0,\perp,\perp)$.

(A.2)

Let

$$\begin{aligned} &\mathsf{frk}\\ &= \Pr\left[b = 1 : x \longleftarrow \mathcal{I}; (b,\sigma^*,\sigma^{**}) \longleftarrow \mathcal{F}_{\mathcal{A}}(x)\right]. \end{aligned} \tag{A.3}$$

Then

$$\mathsf{frk} \geq \mathsf{acc}\cdot\left(\frac{\mathsf{acc}}{q} - \frac{1}{h}\right). \tag{A.4}$$

Alternatively,

$$\mathsf{acc} \leq \frac{q}{h} + \sqrt{q\cdot \mathsf{frk}}. \tag{A.5}$$

B. Security Proofs for the Construction in Section 5

B.1. Proof of Theorem 6. We prove the theorem for the case that message-dependent tokens are issued at most k times. The theorem for the case of an unbounded number of tokens can be proved similarly.

Proof. Let $\mathcal{A}$ be an adversary against the opener anonymity of the proposed scheme. The proof proceeds with a sequence of games.

(i) *Game 0*: This is identical to the opener anonymity game of the proposed scheme.

(ii) *Game 1*: In this game the common reference string σ in *par* is generated by S_1, and the zero-knowledge proof π in the challenge signature σ^* is generated by S_2.

(iii) *Game 2*: In this game χ is sampled randomly and independently.

Let S_i be the event that $\mathcal{A}$ wins in Game i. From the triangle inequality, we have that

$$\begin{aligned} \left|\Pr[S_0] - \frac{1}{2}\right| \leq \sum_{i=0}^{1}\left|\Pr[S_i] - \Pr[S_{i+1}]\right|\\ + \left|\Pr[S_2] - \frac{1}{2}\right|. \end{aligned} \tag{B.1}$$

We then bound each term in this equation.

Lemma B.1. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible, provided that the non-interactive proof system is adaptively zero-knowledge.*

Proof (of Lemma B.1). Using $\mathcal{A}$ we construct an algorithm $\mathcal{B}$ against the zero-knowledge property of the proof system. The construction is as follows.

(i) *Setup*: $\mathcal{B}$ receives a common reference string σ. Then $\mathcal{B}$ sets up gpk, ok, ak, and $(gsk_i)_{i\in[n]}$ except that for the common reference string in gpk $\mathcal{B}$ uses the common reference string σ received from its challenger. Then $\mathcal{B}$ sends $(gpk, ok, (gsk_i)_{i\in[n]})$ to $\mathcal{A}$.

(ii) *Token Query (Phase I)*: When $\mathcal{A}$ issues a token query M, $\mathcal{B}$ responds with t_M generated with ak.

(iii) *Challenge*: When $\mathcal{A}$ request a challenge for i_0, i_1, and M^*, $\mathcal{B}$ chooses a random bit μ. Using gsk_{i_μ}, $\mathcal{B}$ generate a challenge signature as in the opener anonymity game with an exception that the proof π in the challenge signature is obtained by querying the statement and the witness to $\mathcal{B}$'s challenger.

(iv) *Token Query (Phase II)*: Token queries from $\mathcal{A}$ are responded as in Phase I.

(v) *Guess*: When $\mathcal{A}$ outputs a guess μ', $\mathcal{B}$ outputs 1 if $\mu = \mu'$ and outputs 0 otherwise.

Observe that if $\mathcal{B}$ receives a proof generated by P, $\mathcal{B}$ simulates Game 0 perfectly. If $\mathcal{B}$ receives a proof generated by S_2, $\mathcal{B}$ simulates Game 1 perfectly. Then we have that $|\Pr[S_0] - \Pr[S_1]| = 2\mathrm{Adv}^{\mathrm{zk}}_{\mathcal{B}}(\lambda)$, which is negligible. □

Lemma B.2. *$|\Pr[S_1] - \Pr[S_2]|$ is negligible, provided that the identity-based KEM is k-resilient.*

Proof (of Lemma B.2). Using $\mathcal{A}$ we construct an algorithm $\mathcal{B}$ against the k-resilience of the identity-based KEM. The construction of $\mathcal{B}$ is as follows.

(i) *Setup*: $\mathcal{B}$ receives a public parameter par. Then $\mathcal{B}$ sets up gpk, ok, $(gsk)_{i\in[n]}$ following the description of Game 1 with an exception that for the public parameter in gpk $\mathcal{B}$ uses par sent from its challenger. Then $\mathcal{B}$ sends $(gpk, ok, (gsk_{i\in[n]})_i)$ to $\mathcal{A}$.

(ii) *Token Query (Phase I)*: When $\mathcal{A}$ issues a token query M, $\mathcal{B}$ issues an extraction query M to its challenger and receives a decryption key dk_M. Then $\mathcal{B}$ returns $t_M = dk_M$ to $\mathcal{A}$.

(iii) *Challenge*: When $\mathcal{A}$ requests a challenge for i_0, i_1, and M^*, $\mathcal{B}$ chooses a random bit μ. Then $\mathcal{B}$ requests a challenge (C^*, K^*) to its challenger under the identity M^*. Using these C^* and K^* as C_{IBE} and K_{IBE}, $\mathcal{B}$ generates a challenge signature σ^* as in Game 1. Notice that due to the change in Game 1, in order to generate a proof π, $\mathcal{B}$ no longer needs to know the randomness behind C_{IBE}. Then $\mathcal{B}$ returns a challenge signature σ^* to $\mathcal{A}$.

(iv) *Token Query (Phase II)*: In this phase, token queries from $\mathcal{A}$ are responded as in Phase I.

(v) *Guess*: When $\mathcal{A}$ outputs a guess μ', $\mathcal{B}$ outputs 1 if $\mu = \mu'$, and outputs 0 otherwise.

Observe that if $\mathcal{B}$ receives a session key $K^* = K_0$ (encapsulated in C^*), $\mathcal{B}$ perfectly simulates Game 1. Furthermore, if $\mathcal{B}$ receives a session key $K^* = K_1$ (independently sampled), $\mathcal{B}$ perfectly simulates Game 2. This is because K_1 is independent of any other values, hence χ is distributed randomly and independently. Furthermore, $\mathcal{A}$ issues at most k token queries and does not issue M^* as a token query. Therefore, we have that $|\Pr[S_1] - \Pr[S_2]| = 2\mathrm{Adv}^{\mathrm{tb-KEM}}_{\mathcal{B}}(\lambda)$, which is negligible. □

Lemma B.3. $|\Pr[S_2] - 1/2| = 0$.

Proof (of Lemma B.3). Consider the process of generating the challenge signature in Game 2. In this process, the only variable that directly depends on the challenge bit μ is $s \longleftarrow \mathsf{Sign}(sk_{i_\mu}, M)$. However this s is no longer used to generate a challenge signature any more, and thus the challenge signature is independent of the challenge bit μ. Hence the lemma holds. □

Finally, we have that all the three terms in Eq. (B.1) are negligible, hence the advantage of $\mathcal{A}$ in the opener anonymity game is negligible.

This complete the proof of Theorem 6. □

B.2. Proof of Theorem 7

Proof. Let $\mathcal{A}$ be an admitter anonymity adversary against the proposed scheme. The proof proceeds with a sequence of games. We define the following four games.

(i) *Game 0*. The initial game is identical to the game defined in the definition of admitter anonymity. For the subsequent games, vk^*_{OT} is generated before running the adversary $\mathcal{A}$.

(ii) *Game 1*. In this game, an opening query which contains a valid one-time signature with $vk = vk^*$ is responded with $\perp$.

(iii) *Game 2*. In this game, we replace the common reference string with a simulated string generated by S_1 and the NIZK proof, included in challenge, with a simulated proof using S_2 of the NIZK proof system.

(iv) *Game 3*. In this game, we change χ^* of the challenge signature to be a random element.

For $i = 0, 1, 2, 3$, we define the following two events: Let S_i denote the event that the adversary $\mathcal{A}$ successfully guesses the challenge bit in Game i. Let F_i denote the event that $\mathcal{A}$ submits an open query $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ in Game i where $vk_{\mathrm{OT}} = vk^*_{\mathrm{OT}}$ and $\mathsf{Verify}^{\mathrm{OT}}(vk_{\mathrm{OT}}, \langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}}) = \top$.

The advantage of $\mathcal{A}$ is $|\Pr[S_0] - 1/2|$ from the definition. From the triangle inequality, the following inequality holds.

$$\left|\Pr[S_0] - \frac{1}{2}\right| \le \sum_{i=0}^{2} \left|\Pr[S_i] - \Pr[S_{i+1}]\right| + \left|\Pr[S_3] - \frac{1}{2}\right|. \tag{B.2}$$

In what follows, we prove that each term of the above inequality is negligible.

Lemma B.4. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible if the underlying one-time signature scheme is strongly unforgeable.*

Proof (of Lemma B.4). By the difference lemma [67], Game 0 and Game 1 are equivalent if the event F_i does not occur. Therefore,

$$\left|\Pr\left[S_0\right] - \Pr\left[S_1\right]\right| \leq \Pr\left[F_0\right] = \Pr\left[F_1\right]. \tag{B.3}$$

We then prove $\Pr[F_0]$ $(= \Pr[F_1])$ is negligible.

To prove $\Pr[F_0]$ is negligible, we will construct another adversary $\mathcal{F}$, which attacks strong unforgeability of the one-time signature scheme, and relate its success probability with the probability of the event F_0. The construction of $\mathcal{F}$ is as follows:

(i) *Setup*: The adversary $\mathcal{F}$ first receives a verification key vk^*_{OT} for the one-time signature scheme. The adversary then runs $\mathsf{GKg}(1^\lambda, 1^n, 1^k)$ to obtain a group public key $gpk = (vk_{\mathrm{issue}}, pk, par, \Sigma)$, the opening key ok, the admitting key ak, and user signing keys $gsk_i = (i, vk_i, cert_i, sk_i)$ for all $i \in [n]$. Then $\mathcal{B}$ runs $\mathcal{A}(gpk, ak, (gsk_i)_{i\in[n]})$.

(ii) *Open Query (Phase I)*: Queries are answered with the opening key ok and the admitting key ak. In addition, when $\mathcal{A}$ queries a group signature $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ in which $vk_{\mathrm{OT}} = vk^*_{\mathrm{OT}}$ and $\mathsf{Verify}(vk_{\mathrm{OT}}, \langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}}) = \top$, $\mathcal{F}$ records the pair $(\langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}})$ and responds with $\perp$.

(iii) *Challenge*: To respond to the challenge request (i_0, i_1, M^*), $\mathcal{F}$ chooses a random bit b and generates a group signature $(vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$ in exactly the same way as the construction with one exception that σ^*_{OT} is obtained by issuing a signing query $\langle C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*\rangle$ to the challenger.

(iv) *Open Query (Phase II)*: Further open queries are answered as in Phase I.

(v) *Guess*: When $\mathcal{A}$ outputs a guess and halts, if there is a recorded tuple, $\mathcal{F}$ outputs this tuple as a forgery. Otherwise $\mathcal{F}$ outputs $(\perp, \perp)$.

This adversary $\mathcal{F}$ perfectly simulates Game 0 (or Game 1) until the event F_0 (or F_1) occurs. Furthermore, whenever the event F_0 happens, this adversary $\mathcal{F}$ successfully outputs a forgery and wins the game (because $(C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ must be different from $(C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$, and it consists of a legitimate forgery). Then we can conclude $\Pr[F_0]$ is negligible, because of the security of the underlying one-time signature scheme. This completes the proof of Lemma B.4. □

Lemma B.5. $|\Pr[S_1] - \Pr[S_2]|$ *is negligible if the underlying non-interactive proof system is adaptively zero-knowledge.*

It is straightforward to prove that the difference between $\Pr[S_1]$ and $\Pr[S_2]$ is negligible from the adaptive zero-knowledge property of the NIZK proof system, hence we omit the detailed proof of Lemma B.5.

Lemma B.6. $|\Pr[S_2] - \Pr[S_3]|$ *is negligible if the underlying tag-based KEM is selective-tag weakly chosen-ciphertext secure.*

Proof (of Lemma B.6). We will construct an adversary $\mathcal{B}$ which attacks the underlying tag-based KEM. The construction of $\mathcal{B}$ is as follows:

(i) *Setup*: The adversary $\mathcal{B}$ first runs $\mathsf{SigKg}^{\mathrm{OT}}(1^\lambda)$ to generate a verification/signing key pair $(vk^*_{\mathrm{OT}}, sk^*_{\mathrm{OT}})$, outputs vk^*_{OT} as the challenge tag, and then receives the public key pk of the tag-based KEM. The adversary $\mathcal{B}$ then generates the rest of a group public key as $(vk_{\mathrm{issue}}, sk_{\mathrm{issue}}) \longleftarrow \mathsf{SigKg}(1^\lambda)$, $(par, mk) \longleftarrow \mathsf{ISetup}(1^\lambda)$, $(\Sigma, \tau) \longleftarrow S_1(1^\lambda)$, user signing keys $(vk_i, sk_i) \longleftarrow \mathsf{SigKg}^{\mathrm{OT}}(1^\lambda)$ for all $i \in [n]$, and their certificates $cert_i \longleftarrow \mathsf{Sign}(sk_{\mathrm{issue}}, \langle i, vk_i\rangle)$ for all $i \in [n]$. The adversary $\mathcal{B}$ then sets $gpk \longleftarrow (vk_{\mathrm{issue}}, pk, par, \Sigma)$ and $gsk_i \longleftarrow (i, vk_i, cert_i, sk_i)$ for all $i \in [n]$ and runs $\mathcal{A}$ with the input $(gpk, mk, (gsk_i)_{i\in[n]})$.

(ii) *Open Query (Phase I)*: When the adversary $\mathcal{A}$ submits an open query for a signature $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi, \sigma_{\mathrm{OT}})$ and a message M, the adversary $\mathcal{B}$ responds as follows: (i) when $vk_{\mathrm{OT}} \neq vk^*_{\mathrm{OT}}$, $\mathcal{B}$ makes a decapsulation query for the ciphertext C_{PKE} with a tag vk_{OT} to obtain a session key K_{PKE} (note that this query is legitimate), and then extracts a user decryption key dk_M (of the identity-based KEM) from mk by running $dk_M \longleftarrow \mathsf{IExt}(par, mk, M)$, decrypts C_{IBE} with dk_M to obtain a session key K_{IBE} by running $K_{\mathrm{IBE}} \longleftarrow \mathsf{IDec}(dk_M, M, C_{\mathrm{IBE}})$, and verifies whether $\mathsf{Verify}^{\mathrm{OT}}(vk_{\mathrm{OT}}, \langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}}) = \top$ and $V_{\mathrm{NIZK}}(\Sigma, (vk_{\mathrm{issue}}, pk, par, C_{\mathrm{PEK}}, C_{\mathrm{IBE}}), \pi) = \top$ hold. If both of them hold, $\mathcal{B}$ further computes $\langle i, vk, cert, s\rangle \longleftarrow \chi \odot K^{-1}_{\mathrm{IBE}} \odot K^{-1}_{\mathrm{PKE}}$ and responds with i. Otherwise $\mathcal{B}$ responds with $\perp$. (ii) When $vk_{\mathrm{OT}} = vk^*_{\mathrm{OT}}$, $\mathcal{B}$ responds with $\perp$.

(iii) *Challenge*: At some time $\mathcal{A}$ requests a challenge for (i_0, i_1, M^*), then $\mathcal{B}$ computes a challenge as follows: $\mathcal{B}$ generates a signature $s_\mu \longleftarrow \mathsf{Sign}(sk_{i_\mu}, M^*)$ for a random bit $\mu \longleftarrow \{0, 1\}$, requests a challenge to obtain (C^*, K^*), generates a ciphertext and a session key as $(C_{\mathrm{IBE}}, K_{\mathrm{IBE}}) \longleftarrow \mathsf{IEnc}(par, M^*)$, computes $\chi \longleftarrow \langle i_\mu, vk_{i_\mu}, cert_{i_\mu}, s_{i_\mu}\rangle \odot K^* \odot K_{\mathrm{IBE}}$, and generates a fake proof π by computing $\pi \longleftarrow S_2(\Sigma, (vk_{\mathrm{issue}}, pk, par, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi), \tau)$. Finally $\mathcal{B}$ signs $\langle vk^*_{\mathrm{OT}}, C^*, C_{\mathrm{IBE}}, \chi, \pi\rangle$ with the one-time signing key sk^*_{OT} to obtain σ^*_{OT} and sends $(vk^*_{\mathrm{OT}}, C^*, C_{\mathrm{IBE}}, \chi, \pi, \sigma^*_{\mathrm{OT}})$ to $\mathcal{A}$.

(iv) *Open Query (Phase II)*: Again $\mathcal{A}$ submits more open queries and $\mathcal{B}$ responds as before.

(v) *Guess*: When $\mathcal{A}$ outputs a bit μ', $\mathcal{B}$ outputs 1 if $\mu' = \mu$, otherwise outputs 0.

Let $\mathrm{Adv}^{\mathrm{tb-KEM}}_{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$.

The adversary $\mathcal{B}$ perfectly simulates Game 2 and Game 3 when K^*_{PKE} is the real key and a random value, respectively,

so that $\mathcal{A}$'s challenge bit is μ. Therefore, $\mathcal{B}$ outputs 1 when $\mathcal{A}$ successfully guesses the challenge bit in each game.

Hence, we have that $|\Pr[S_2] - \Pr[S_3]| = 2\mathrm{Adv}^{\mathrm{tb-KEM}}_{\mathcal{B}}(\lambda)$. When the tag-based KEM is secure, $\mathrm{Adv}^{\mathrm{tb-KEM}}_{\mathcal{B}}(\lambda)$ is negligible and therefore $|\Pr[S_2] - \Pr[S_3]|$ is also negligible. □

Lemma B.7. $|\Pr[S_3] - 1/2| = 0$.

Proof (of Lemma B.7). In Game 3, χ^* is a random value. That is, $\mathcal{A}$'s view is completely independent of the challenge bit b in Game 3, and therefore the challenge signature does not contain the information of b. □

From the above, the proof of Theorem 7 is completed. □

B.3. Proof of Theorem 8

Proof. Let $\mathcal{A}$ be a traceability adversary against the proposed scheme. Let S be the event that $\mathcal{A}$ wins the traceability game. Let (M^*, σ^*) be the output of $\mathcal{A}$. Furthermore we denote σ^* as $(vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*, \sigma^*_{\mathrm{OT}})$.

We define the following events:

(i) S: The adversary $\mathcal{A}$ satisfies the winning condition.

(ii) P: The statement that π^* proves is in the language.

Let $\langle i^*, vk^*, cert^*, s^*\rangle$ be $\chi^* \odot {K^*_{\mathrm{IBE}}}^{-1} \odot {K^*_{\mathrm{PKE}}}^{-1}$ where K^*_{IBE} and K^*_{PKE} be the decapsulated session key computed in the Open algorithm. Let $(i, vk_i, cert_i, sk_i)$ be the user i's signing key generated in the game. We further define the following events:

(i) $C_{(\mathrm{i})}$: The event P occurs and the adversary $\mathcal{A}$ wins the game satisfying the winning condition *(i)*.

(ii) $C_{(\mathrm{ii})}$: The event P occurs and the adversary $\mathcal{A}$ wins the game satisfying the winning condition *(ii)*.

(a) $C_{(\mathrm{ii}),1}$: The event $C_{(\mathrm{ii})}$ occurs and none of $i \in [n]$ satisfies $(i^*, vk^*) = (i, vk_i)$.

(b) $C_{(\mathrm{ii}),2}$: The event $C_{(\mathrm{ii})}$ occurs and it holds that $(i^*, vk^*) = (i, vk_i)$ for some $i \in [n]$.

Then we have that

$$S = (S \wedge \neg P) \vee C_{(\mathrm{i})} \vee C_{(\mathrm{ii}),1} \vee C_{(\mathrm{ii}),2}. \tag{B.4}$$

Since $S \wedge \neg P$, $C_{(\mathrm{i})}$, $C_{(\mathrm{ii}),1}$, and $C_{(\mathrm{ii}),2}$ are exclusive, we have that

$$\begin{aligned}\Pr[S] = {} & \Pr[S \wedge \neg P] + \Pr\left[C_{(i)} \vee C_{(\mathrm{ii}),1}\right] \\ & + \Pr\left[C_{(\mathrm{ii}),2}\right].\end{aligned} \tag{B.5}$$

We then bound each term.

Lemma B.8. $\Pr[S \wedge \neg P]$ *is negligible provided that the proof system is adaptively sound.*

Proof. Given the adversary $\mathcal{A}$ we construct an algorithm $\mathcal{B}$ that attacks the soundness of the proof system. The construction of $\mathcal{B}$ is as follows: $\mathcal{B}$ receives as an input a common reference string; then $\mathcal{B}$ sets up a group public key, an opening key, an admitting key, and signing keys; $\mathcal{B}$ executes the traceability game using these keys; when $\mathcal{A}$ terminates with output (M^*, σ^*) where σ^* is parsed as $(vk_{\mathrm{OT}}, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \xi, \pi, \sigma_{\mathrm{OT}})$ $\mathcal{B}$ outputs π together with the statement.

Whenever the event $S \wedge \neg P$ occurs, $\mathcal{B}$ successfully outputs a statement/proof pair which is verified as valid but the statement is outside of the language. Hence, since the assumption that the proof system is sound, $\Pr[S \wedge \neg P]$ is negligible. □

Lemma B.9. $\Pr[C_{(\mathrm{i})} \vee C_{(\mathrm{ii}),1}]$ *is negligible provided that the signature scheme is EUF-CMA secure.*

Proof. Given the adversary $\mathcal{A}$ we construct an algorithm $\mathcal{B}$ that attacks the unforgeability of the signature scheme. The construction of $\mathcal{B}$ is as follows: $\mathcal{B}$ receives as an input a verification key vk of the signature scheme; using this vk as vk_{issue}, it sets up a group public key, an opening key, and signing keys; for generating the signing key for the user i, $\mathcal{B}$ issues a signing query for the message $\langle i, vk_i\rangle$ to obtain a signature on that message, and uses this signature as $cert_i$ in the signing key of the user i; $\mathcal{B}$ then runs $\mathcal{A}$ with the input the group public key, opening key, and admitting key; all the key revealing queries and signing queries are responded with the signing keys for the users generated by $\mathcal{B}$; when $\mathcal{A}$ outputs a pair (M^*, σ^*), $\mathcal{B}$ parses σ^* as $(vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}}, C^*_{\mathrm{IBE}}, \chi^*, \pi^*)$, decrypts χ and parses the plaintext by running $\langle i^*, vk^*, cert^*, s\rangle \longleftarrow \chi \odot \mathsf{IDec}(t_{M^*}, M^*, C^*_{\mathrm{IBE}}) \odot \mathsf{TDec}(ok, vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}})$, and finally outputs $(\langle i^*, vk^*\rangle, cert^*)$ as a forgery; if either $\mathsf{IDec}(t_{M^*}, M^*, C^*_{\mathrm{IBE}})$ or $\mathsf{TDec}(ok, vk^*_{\mathrm{OT}}, C^*_{\mathrm{PKE}})$ outputs $\perp$, $\mathcal{B}$ outputs $(\perp, \perp)$.

We then argue that whenever $C_{(\mathrm{i})} \vee C_{(\mathrm{ii}),1}$ occurs, $\mathcal{B}$ successfully outputs a forgery for the signature scheme.

Let us assume $C_{(\mathrm{i})}$ occurs. There are five possibilities that Open outputs $\perp$:

(i) $\mathsf{TDec}(ok, vk^*_{\mathrm{OT}}, C_{\mathrm{PKE}}) = \perp$,

(ii) $\mathsf{IDec}(t_{M^*}, M^*, vk^*_{\mathrm{OT}}, C_{\mathrm{IBE}}) = \perp$,

(iii) $\mathsf{Verify}^{\mathrm{OT}}(vk_{\mathrm{OT}}, \langle C_{\mathrm{PKE}}, C_{\mathrm{IBE}}, \chi, \pi\rangle, \sigma_{\mathrm{OT}}) = \perp$,

(iv) $V_{\mathrm{NIZK}}(\Sigma, (pk_{\mathrm{issue}}, pk, par, C_{\mathrm{PKE}}, C_{\mathrm{IBE}}), \pi) = \perp$, or

(v) $i^* \notin [n]$.

The first two possibilities in fact never occur, since the statement π proves is in the language, and then the correctness condition ensures that these two ciphertexts are correctly decrypted. The third and fourth possibilities never occur, since the winning condition requires that σ^* be verified as valid by GVf, in which σ^*_{OT} and π^* are verified. Therefore whenever $C_{(\mathrm{i})}$ occurs, we have that $i^* \notin [n]$. Since the condition that P occurs ensures that $(\langle i^*, vk^*\rangle, cert^*)$ is verified as valid under vk_{issue}, and $\mathcal{B}$ only issues signing queries of the form $\langle i, vk\rangle$ for some $i \in [n]$ and some vk, the output of $\mathcal{B}$ is a legitimate forgery for vk_{issue}.

Then let us assume $C_{(\mathrm{ii}),1}$ occurs. In this case the event P ensures that $(\langle i^*, vk^*\rangle, cert^*)$ is a valid message-signature pair under vk_{issue}. Furthermore, the condition $C_{(\mathrm{ii}),1}$ ensures

that $i^* \in [n]$ but $vk^* \neq vk_{i^*}$. Thus, the output of $\mathcal{B}$ is again a legitimate forgery for vk_{issue}.

Therefore, whenever $C_{(\text{i})} \vee C_{(\text{ii}),1}$ occurs, $\mathcal{B}$ successfully outputs a forgery. Hence the unforgeability of the signature scheme ensures that $\Pr[C_{(\text{i})} \vee C_{(\text{ii}),1}]$ is negligible. □

Lemma B.10. $\Pr[C_{(\text{ii}),2}]$ *is negligible provided the signature scheme is EUF-CMA secure.*

Proof. Given the adversary $\mathcal{A}$ we construct an algorithm that attacks the unforgeability of the signature scheme. The construction of $\mathcal{B}$ is as follows: $\mathcal{B}$ receives as an input a verification key vk of the underlying signature scheme; $\mathcal{B}$ chooses a random index $j \longleftarrow [n]$ of a signer and sets $vk_j = vk$; $\mathcal{B}$ then generates a group public key, an opening key, an admitting key, and signing keys for $i \in [n] \setminus \{j\}$. $\mathcal{B}$ runs $\mathcal{A}$ with the input the group public key, the opening key, and the admitting key. Key revealing queries $i \neq j$ are responded as in the traceability game, and key revealing queries j are responded with $\perp$. Signing queries (i, M) where $i \neq j$ are responded using the signing keys generated by $\mathcal{B}$ itself. Signing queries (j, M) for any M are responded by querying M as $\mathcal{B}$'s signing query, receiving s as its reply, using s to generate a group signature σ, and responding with this σ; when $\mathcal{A}$ outputs a forgery (M^*, σ^*), $\mathcal{B}$ decrypts χ by running $\langle i^*, vk^*, cert^*, s^* \rangle \longleftarrow \chi \odot \mathsf{IDec}(t_{M^*}, M^*, C^*_{\text{IBE}}) \odot \mathsf{TDec}(ok, vk^*_{\text{OT}}, C_{\text{PKE}})$, confirms that $(i^*, vk^*) = (j, vk_j)$, where vk_j is the part of the user j's signing key. If $(i^*, vk^*) = (j, vk_j)$ then $\mathcal{B}$ outputs (M^*, s^*), and otherwise outputs $(\perp, \perp)$.

We then bound the probability $\Pr[C_{(\text{ii})}]$ by the probability that $\mathcal{B}$ produces a forgery.

Towards this end we argue that whenever the event $C_{(\text{ii}),2} \wedge (i^* = j)$ occurs, $\mathcal{B}$ successfully outputs a forgery for the signature scheme. Let us assume $C_{(\text{ii}),2} \wedge (i^* = j)$ occurs. The event $C_{(\text{ii}),2}$ ensures that (M^*, s^*) is a valid message-signature pair under vk^*. Furthermore, the event $C_{(\text{ii}),2}$ ensures that $vk^* = vk^{i^*}$, where vk_{i^*} is the verification key included in the user i^*'s signing key. Moreover, the event $i^* = j$ ensures that vk_{i^*} is in fact vk, which is the verification key that is given to $\mathcal{B}$. In addition, the winning condition of the traceability game ensures that $\mathcal{A}$ did not issue the signing query (i^*, M^*), which in turn ensures that $\mathcal{B}$ did not issue the signing query M^*. Therefore, we can conclude that whenever $C_{(\text{ii}),2} \wedge (i^* = j)$ occurs, $\mathcal{B}$ successfully outputs a legitimate forgery.

Since i^* is information-theoretically hidden from $\mathcal{A}$, we have

$$\Pr\left[C_{(\text{ii}),2} \wedge \left(i^* = j\right)\right] = \frac{1}{n} \Pr\left[C_{(\text{ii}),2}\right]. \tag{B.6}$$

Then we have that

$$\Pr\left[C_{(\text{ii}),2}\right] \leq n \Pr\left[\mathcal{B} \text{ forges}\right], \tag{B.7}$$

which is negligible. □

Finally we have that all the three terms in Eq. (B.5) are negligible, thus $\Pr[S]$ is negligible.

This complete the proof of Theorem 8. □

C. Building Blocks and Their Security Proofs

C.1. ***k****-Resilient Identity-Based KEM from the DLIN Assumption.* In the following presentation of the identity-based KEM, k denotes the upper bound for the number of the corrupted users, while l denotes the message length (in group elements) to be encrypted. Our proposed k-resilient identity-based KEM based on the Heng-Kurosawa scheme is shown in Box 4.

Theorem C.1. *The construction in Box 4 is a k-resilient identity-based KEM if the DLIN assumption holds.*

Proof. Given an adversary $\mathcal{A}$ which attacks the scheme in Box 4 , we bound its advantage by constructing the reduction $\mathcal{B}$ below:

(i) *Setup*. The simulator $\mathcal{B}$ receives an instance $(u, v, h, u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}})$ of the DLIN problem, where $\tilde{\tilde{r}}$ is either $r + \tilde{r}$ or an independently chosen random element of $\mathbb{Z}_p \setminus \{r + \tilde{r}\}$. $\mathcal{B}$ generates random polynomials $(d_i(x) = d_{i,0} + \cdots + \alpha_{i,k}x^k,\ d'_i(x) = d'_{i,0} + \cdots + d'_{i,k}x^k,\ d''_i(x) = d''_{i,0} + \cdots + d''_{i,k}x^k)_{i\in[l]}$ of degree k, sets $D_{i,j} \longleftarrow u^{d_{i,j}} h^{d''_{i,j}}$ and $\widetilde{D}_{i,j} \longleftarrow v^{d'_{i,j}} h^{d''_{i,j}}$ for all $i \in [l]$ and $j \in \{0, \ldots, k\}$, and runs $\mathcal{A}$ with an input $par = (u, v, h, (D_{i,j}, \widetilde{D}_{i,j})_{i\in[l], j\in\{0,\ldots,k\}})$.

(ii) *Query (Phase I)*. When $\mathcal{A}$ queries an identity ID, $\mathcal{B}$ returns $dk_{ID} = (d_i(ID), d'_i(ID), d''_i(ID))_{i\in[l]}$.

(iii) *Challenge*. When $\mathcal{A}$ requests a challenge for an identity ID^*, $\mathcal{B}$ chooses a random bit $b \longleftarrow \{0, 1\}$. Then $\mathcal{B}$ computes $C^* = (u^r, v^{\tilde{r}}, h^{\tilde{\tilde{r}}})$. If $b = 0$, $\mathcal{B}$ computes $K^* = (K^*_1, \ldots, K^*_l) = ((u^r)^{d_1(ID^*)}(v^{\tilde{r}})^{d'_1(ID^*)}(h^{\tilde{\tilde{r}}})^{d''_1(ID^*)}, \ldots, (u^r)^{d_l(ID^*)}(v^{\tilde{r}})^{d'_l(ID^*)}(h^{\tilde{\tilde{r}}})^{d''_l(ID^*)})$. If $b = 1$, $\mathcal{B}$ chooses a random group elements $K^* \longleftarrow \mathbb{G}^l$. These C^* and K^* are given to $\mathcal{A}$ as a challenge.

(iv) *Query (Phase II)*. Again, $\mathcal{A}$ may request a decryption key for ID and $\mathcal{B}$ responds as before.

(v) *Guess*. Finally $\mathcal{A}$ outputs a bit b' and $\mathcal{B}$ outputs 1 if $b = b'$. Otherwise $\mathcal{B}$ outputs 0.

Let us assume $\tilde{\tilde{r}} = r + \tilde{r}$. Since $\mathcal{B}$ generates the master secret key by itself, and the random exponents r and $\tilde{r}$ in the instance correspond to the random exponents of the encapsulation algorithm, the reduction perfectly simulates the experiment. Otherwise when $\tilde{\tilde{r}} \neq r + \tilde{r}$ and $b = 0$, we will show that K^* is distributed uniformly and independently from all other values seen by $\mathcal{A}$. To see this, let $ID_1, \ldots, ID_k$ be the decapsulation key queries issued by $\mathcal{A}$ during the simulation, and observe that the queries reveal the values $d_i(ID_j)$, $d'_i(ID_j)$, $d''_i(ID_j)$ to $\mathcal{A}$, but $d_i(ID^*)$, $d'_i(ID^*)$, and $d''_i(ID^*)$ are not revealed. However, par further reveals the value $d_i(ID^*) + \alpha d''_i(ID^*)$ and $d'_i(ID^*) + \beta d''_i(ID^*)$, where $u = g^\alpha$ and $v = g^\beta$. The equations $\mathcal{A}$ can observe are represented as in (C.1) where this matrix is nonsingular, and hence K^*

ISetup $(1^\lambda, 1^k)$:
  $(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathcal{G}(1^\lambda)$
  $u, v, h \longleftarrow \mathbb{G}$
  For $i \in [l]$, $j \in \{0, \ldots, k\}$:
    $d_{i,j}, d'_{i,j}, d''_{i,j} \longleftarrow \mathbb{Z}_p$
  For $i \in [l]$, $j \in [k]$:
    $D_{i,j} \longleftarrow u^{d_{i,j}} h^{d''_{i,j}}$
    $\widetilde{D}_{i,j} \longleftarrow v^{d'_{i,j}} h^{d''_{i,j}}$
  For $i \in [l]$:
    $d_i(X) \longleftarrow d_{i,0} + d_{i,1}X + \cdots + d_{i,k}X^k$
    $d'_i(X) \longleftarrow d'_{i,0} + d'_{i,1}X + \cdots + d'_{i,k}X^k$
    $d''_i(X) \longleftarrow d''_{i,0} + d''_{i,1}X + \cdots + d''_{i,k}X^k$
  $par \longleftarrow (p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, (D_{i,j}, \widetilde{D}_{i,j})_{i\in[l], j\in\{0,\ldots,k\}})$
  $mk \longleftarrow (d_i(X), d'_i(X), d''_i(X))_{i\in[l]}$
  Output (par, mk)
IExt (par, mk, ID):
  $dk_{ID} \longleftarrow (d_i(ID), d'_i(ID), d''_i(ID))_{i\in[l]}$
  Output dk_{ID}
IEnc (par, ID):
  $\rho, \widetilde{\rho} \longleftarrow \mathbb{Z}_p$
  $C_{\mathrm{IBE}} \longleftarrow (u^\rho, v^{\widetilde{\rho}}, h^{\rho+\widetilde{\rho}})$
  $K_{\mathrm{IBE}} \longleftarrow ((\prod_{j=0}^k D_{i,j}^{ID^j})^\rho (\prod_{j=0}^k \widetilde{D}_{i,j}^{ID^j})^{\widetilde{\rho}})_{i\in[l]}$
IDec $(dk_{ID}, ID, C_{\mathrm{IBE}})$
  $(d_i(ID), d'_i(ID), d''_i(ID))_{i\in[l]} \longleftarrow dk_{ID}$
  $(C_1, C_2, C_3) \longleftarrow C_{\mathrm{IBE}}$
  Output $(C_1^{d_i(ID)} C_2^{d'_i(ID)} C_3^{d''_i(ID)})_{i\in[l]}$

Box 4: Our k-resilient IBE scheme, in which l denotes the message length (in group elements) of the scheme.

is uniformly distributed. This distribution is identical to the case of $b = 1$, hence b is independent of $\mathcal{A}$'s view. Therefore, assuming the DLIN assumption, we have that

$$\begin{pmatrix} \left(dk_{ID_1}\right) \\ \vdots \\ \left(dk_{ID_k}\right) \\ \log_g u^{x_i(ID^*)} h^{x''_i(ID^*)} \\ \log_g v^{x'_i(ID^*)} h^{x''_i(ID^*)} \\ \log_g K_i^* \end{pmatrix} = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & \alpha & 1 \\ & & & & & & & \beta & 1 \\ & & & & & & r\alpha & \widetilde{r}\beta & \widetilde{\widetilde{r}} \end{pmatrix} \begin{pmatrix} d_i(ID_1) \\ d'_i(ID_1) \\ d''_i(ID_1) \\ \vdots \\ d_i(ID_k) \\ d'_i(ID_k) \\ d''_i(ID_k) \\ d_i(ID^*) \\ d'_i(ID^*) \\ d''_i(ID^*) \end{pmatrix}, \tag{C.1}$$

$$\begin{aligned} &\left|\Pr\left[b = b' \mid r + \widetilde{r} = \widetilde{\widetilde{r}}\right] - \Pr\left[b = b' \mid r + \widetilde{r} \neq \widetilde{\widetilde{r}}\right]\right| \\ &= \left|\Pr\left[b = b' \mid r + \widetilde{r} = \widetilde{\widetilde{r}}\right] - \frac{1}{2}\right| \end{aligned} \tag{C.2}$$

is negligible. This is identical to the advantage of $\mathcal{A}$, hence the theorem is proven. □

C.2. Abe-Haralambiev-Ohkubo Signature. The Abe-Haralambiev-Ohkubo signature scheme, which is a structure-preserving signature scheme based on the SFP assumption [6, 60] is shown in Box 5 In the box, the Rand algorithm is given a pair of group elements (g, h) and produces the pair uniformly random under the constraint that the pairing of the new pair is unchanged from $e(g, h)$. The Extend algorithm is given a pair (g, h) and produces a set of pairs (u, u') and (v, v') which is uniformly distributed under the constraint that $e(g, h) = e(u, u')e(v, v')$.

The security of the scheme is as follows.

Theorem C.2. *The construction in Box 5 is EUF-CMA secure if the SFP assumption holds.*

C.3. Shacham's Variant of Cramer-Shoup Encryption. Shacham [13] proposed a variant of the Cramer-Shoup encryption scheme [12, 65] modified to be based on the DLIN assumption. The scheme below further modifies the Shacham's variants in two points: (1) Used as a tag-based KEM and (2) modified to encapsulate a long session key in a constant-size ciphertext. We omit the proof that this scheme is secure under the DLIN assumption, which can be easily obtained by modifying the original proof by Shacham [13]. This modified Shacham's variant is shown in Box 6.

Theorem C.3. *The construction in Box 6 is a selective-tag weakly chosen-ciphertext secure if the DLIN assumption holds.*

D. Security Proofs for the Construction in Section 7

D.1. Proof of Theorem 12

Proof. The proof proceeds with a sequence of games. Let $\mathcal{A}$ be an adversary against the opener anonymity. We consider the following games, which $\mathcal{A}$ plays. We denote by S_i the event in which $\mathcal{A}$ outputs a bit b' which is equal to the challenge bit b flipped by the challenger. We assume that before issuing a token query M $\mathcal{A}$ issues the H_1 query M. We do not lose generality, since given any adversary $\mathcal{A}$ which not necessarily meets this restriction, we can easily modify $\mathcal{A}$ to one that meets this restriction.

(i) *Game 0.* This initial game is identical to the game defined in the description of the opener anonymity. We assume the challenger to maintain two hash lists for H_1 and H_2, which, respectively, contain tuples of the form (M, w, d) and the form $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c)$ to represent $H_1(M) = w$ and $H_2(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}) = c$. The value

SigKg $(1^\lambda, 1^l)$:
  $(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathcal{G}(1^\lambda)$
  $g', h' \longleftarrow \mathbb{G} \setminus \{1\}$
  For $i \in [l]$:
    $\gamma_i, \delta_i \longleftarrow \mathbb{Z}_p^*$
    $g_i \longleftarrow g'^{\gamma_i}$
    $h_i \longleftarrow g'^{\delta_i}$
  $\gamma'', \delta'' \longleftarrow \mathbb{Z}_p^*$
  $g'' \longleftarrow g'^{\gamma''}$
  $h'' \longleftarrow h'^{\delta''}$
  $\alpha \longleftarrow \mathbb{Z}_p^*$
  $((a_0, \tilde{a}_0), (a_1, \tilde{a}_1)) \longleftarrow \mathsf{Extend}(g', g^\alpha)$
  $\beta \longleftarrow \mathbb{Z}_p^*$
  $((b_0, \tilde{b}_0), (b_1, \tilde{b}_1)) \longleftarrow \mathsf{Extend}(g', g^\beta)$
  $vk \longleftarrow (g'', h'', g', h', (g_i, h_i)_{i\in[l]}, a_0, \tilde{a}_0, b_0, \tilde{b}_0, a_1, \tilde{a}_1, b_1, \tilde{b}_1)$
  $sk \longleftarrow (\alpha, \beta, \gamma'', \delta'', (\gamma_i, \delta_i)_{i\in[l]})$
  Output (vk, sk)
Sign $(sk, (m_1, \ldots, m_l))$:
  $\zeta, \rho, \tau, \phi\ \omega \longleftarrow \mathbb{Z}_p$
  $z \longleftarrow g^\zeta$
  $r \longleftarrow g^{\alpha-\rho\tau-\gamma_z\zeta} \prod_{i=1}^l m_i^{-\gamma_i}$
  $s \longleftarrow g'^\rho$
  $t \longleftarrow g^\tau$
  $u \longleftarrow g^{\beta-\phi\omega-\delta_z\zeta} \prod_{i=1}^l m_i^{-\delta_i}$
  $v \longleftarrow h'^\phi$
  $w \longleftarrow g^\omega$
  Output (z, r, s, t, u, v, w)
Verify (vk, m, σ):
  $(m_1, \ldots, m_l) \longleftarrow m$
  $(z, r, s, t, u, v, w) \longleftarrow \sigma$
  If $e(a_0, \tilde{a}_0)e(a_1, \tilde{a}_1) = e(g'', z)e(g', r)e(s, t) \prod_{i=1}^k e(g_i, m_i)$
    and $e(b_0, \tilde{b}_0)e(b_1, \tilde{b}_1) = e(h'', z)e(h', u)e(v, w) \prod_{i=1}^l e(h_i, m_i)$ then
    Output $\top$
  Else
    Output $\perp$
Rand (g, h):
  If $g \neq 1$ and $h \neq 1$ then
    $s \longleftarrow \mathbb{Z}_p^*$
    Output $(g^s, h^{1/s})$
  Else
    $q \longleftarrow \{1, \ldots, 2p-1\}$
    If $q = 1$ then
      Output $(1, 1)$
    Else
      $x \longleftarrow \mathbb{G} \setminus \{1\}$
      $t \longleftarrow \{0, 1\}$
      If $t = 0$ then
        Output $(1, x)$
      Else
        Output $(x, 1)$
Extend (g, h):
  $x \longleftarrow \mathbb{G}$
  $r \longleftarrow \mathbb{Z}_p$
  $U \longleftarrow \mathsf{Rand}(gx^r, h)$
  $V \longleftarrow \mathsf{Rand}(x, h^{-r})$
  Output (U, V)

Box 5: The Abe-Haralambiev-Ohkubo signature scheme, where l denotes the message length (in group elements) of the scheme. The Rand algorithm randomizes a pair (g, h) under the constraint that the pairing $e(g, h)$ is unchanged. The Extend algorithm produces a set of pairs $U = (u, u')$ and $V = (v, v')$ which are uniformly random with the constraint that $e(g, h) = e(u, u')e(v, v')$.

$\mathsf{TKg}\ (1^\lambda)$:
$(p, \mathbb{G}, \mathbb{G}_T, e, g) \longleftarrow \mathcal{G}(1^\lambda)$
$u, v, h \longleftarrow \mathbb{G}$
$x, x', x'', y, y', y'' \longleftarrow \mathbb{Z}_p$
For $i \in [l]$:
$z_i, z_i', z_i'' \longleftarrow \mathbb{Z}_p$
$X \longleftarrow u^x h^{x''}, \widetilde{X} \longleftarrow v^{x'} h^{x''}$
$Y \longleftarrow u^y h^{y''}, \widetilde{Y} \longleftarrow v^{y'} h^{x''}$
For $i \in [l]$:
$Z_i \longleftarrow u^{z_i} h^{z_i''}, \widetilde{Z}_i \longleftarrow h^{z_i''}$
$pk \longleftarrow (u, v, h, X, \widetilde{X}, Y, \widetilde{Y}, (Z_i, \widetilde{Z}_i)_{i\in[l]})$
$dk \longleftarrow (x, x', x'', y, y', y'', (z_i, z_i', z_i'')_{i\in[l]})$
$\mathsf{TEnc}\ (pk, t)$:
$r, \tilde{r} \longleftarrow \mathbb{Z}_p$
$C_{\mathrm{PKE}} \longleftarrow (u^r, v^{\tilde{r}}, h^{r+\tilde{r}}, (XY^t)^r (\widetilde{X}\widetilde{Y}^t)^{\tilde{r}})$
$K_{\mathrm{PKE}} \longleftarrow (Z_1{}^r \widetilde{Z}_1{}^{\tilde{r}}, \ldots, Z_l{}^r \widetilde{Z}_l{}^{\tilde{r}})$
Output $(C_{\mathrm{PKE}}, K_{\mathrm{PKE}})$
$\mathsf{TDec}\ (dk, t, C)$:
$(c_1, c_2, c_3, c_4) \longleftarrow C$
If $c_1{}^{x+ty} c_2{}^{x'+ty'} c_3{}^{x''+ty''} = c_4$ then
Output $(c_1{}^{z_1} c_2{}^{z_1'} c_3{}^{z_1''}, \ldots, c_1{}^{z_l} c_2{}^{z_l'} c_3{}^{z_l''})$

Box 6: Shacham's variant of the Cramer-Shoup encryption scheme, where l is the message length (in group elements) of the scheme.

d is used to answer the random oracle query and the token query in the subsequent games.

(ii) *Game 1.* In this game we replace the zero-knowledge proof of the challenge signature with a simulated proof. More specifically, when the adversary asks a challenge signature $(T_1^*, \ldots, T_6^*, c^*, R_1^*, \ldots, R_{10}^*)$ by sending (i_0, i_1, M), the challenger computes it as follows: the challenger flips the bit $b \in \{0, 1\}$, computes $(T_1^*, \ldots, T_6^*)$ as specified in the construction with the signing key gsk_{i_b}, generates random integers c^*, s_α^*, s_β^*, s_ρ^*, s_η^*, s_x^*, $s_{\alpha x}^*$, $s_{\beta x}^*$, $s_{\rho x}^*$, $s_{\eta x}^* \longleftarrow \mathbb{Z}_p^*$, and computes

$$R_1^* \longleftarrow u^{s_\alpha^*} T_1^{*-c^*},$$

$$R_2^* \longleftarrow v^{s_\beta^*} T_2^{*-c^*},$$

$$R_3^* \longleftarrow h^{s_\alpha^* + s_\beta^*} T_3^{*-c^*},$$

$$R_4^* \longleftarrow e(T_4^*, g)^{s_x^*} e(g_1, w)^{-s_\alpha^*} e(g_1, g)^{-s_{\alpha x}^*} \cdot e(g_2, w)^{-s_\beta^*} e(g_2, g)^{-s_{\beta x}^*} \cdot e(g, w)^{-s_\eta^*} e(g, g)^{-s_{\eta x}^*} \cdot \left(\frac{e(g, g)}{e(T_4^*, w)}\right)^{-c^*},$$

$$R_5^* \longleftarrow g^{s_\rho^*} T_5^{*-c^*},$$

$$R_6^* \longleftarrow e(y, H_1(M))^{s_\rho^*} e(g, g)^{-s_\eta^*} T_6^{*-c^*},$$

$$R_7^* \longleftarrow T_1^{*s_x^*} u^{-s_{\alpha x}^*},$$

$$R_8^* \longleftarrow T_2^{*s_x^*} v^{-s_{\beta x}^*},$$

$$R_9^* \longleftarrow T_5^{*s_x^*} g^{-s_{\rho x}^*},$$

$$R_{10}^* \longleftarrow T_6^{*s_x^*} e(y, H_1(M))^{-s_{\rho x}^*} e(g, g)^{s_{\eta x}^*}. \tag{D.1}$$

The challenger adds the tuple $(M, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c^*)$ to the hash list for H_2. At this point if the list for H_2 already contains a tuple of the form $(M, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c)$ for some c, the challenger outputs $\perp$ and halts. Otherwise the challenger sends $(T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ to $\mathcal{A}$ as the challenge signature. We will argue that this change introduces only a negligible difference in $\mathcal{A}$'s advantage.

(iii) *Game 2.* In this game we change the T_6-component of the challenge signature to be a random element in $\mathbb{G}_T$. This change also introduces a difference of a negligible amount in the $\mathcal{A}$'s advantage.

We then prove that in the last game $\mathcal{A}$ has no information about the bit b, and hence the advantage is zero.

Lemma D.1. $|\Pr[S_2] - 1/2| = 0$.

Proof (of Lemma D.1). In Game 2 $\mathcal{A}$'s view is completely independent of the bit b. The component which is computed using b (and hence has information about b) is T_4^* in the challenge signature. Actually it is distributed independently of b (uniformly over $\mathbb{G}$) due to the random integer η. □

Finally we prove that any of the game-hopping does not change $\mathcal{A}$'s advantage nonnegligibly.

The difference between Game 0 and Game 1 is bounded by a standard argument of the zero-knowledge simulation of the underlying protocol, which is as follows.

Lemma D.2. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible.*

Proof (of Lemma D.2). We claim that the distribution of the challenge in Game 1 is identical to that in Game 0 except for the cases in which the challenger outputs $\perp$. This follows from a standard argument of zero-knowledge simulation. To see this, we can observe that $s_\alpha^* - c^*\alpha$ in Game 1 corresponds to r_α in Game 0, and similar correspondences hold for all the other s^*'s and r's. We can also see that both $s_\alpha^* - c^*\alpha$ and r_α are uniformly distributed over $\mathbb{Z}_p$. We will then see that the challenger in Game 1 outputs $\perp$ only with negligible probability. This is because $(R_1^*, \ldots, R_{10}^*)$ are distributed uniformly over a set with cardinality (at least) p, that is, the queries to H_2 issued before the challenge phase contain $(M, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c)$ with probability (at most) q_{H_2}/p where q_{H_2} denotes the number of oracle queries to H_2 issued by $\mathcal{A}$. □

The difference between Game 1 and Game 2 is bounded by extending the original proof of the BF IBE scheme to that based on the decision assumption.

Lemma D.3. $|\Pr[S_1] - \Pr[S_2]|$ *is negligible under the DBDH assumption.*

Proof (of Lemma D.3). We describe a distinguishing algorithm $\mathcal{B}$ for the DBDH problem to bound the quantity $|\Pr[S_1] - \Pr[S_2]|$. The construction of the algorithm $\mathcal{B}$ is as follows:

(i) *Setup*. $\mathcal{B}$ receives a problem instance $(g^{\delta_1}, g^{\delta_2}, g^{\delta_3}, e(g,g)^{\tau})$, in which τ is either $\delta_1\delta_2\delta_3$ or a random integer δ. $\mathcal{B}$ lets $y = g^{\delta_1}$, and the rest of the components of the group public key gpk and the signing keys gsk_i for all $i \in [n]$ are generated by following the description of the scheme. $\mathcal{B}$ sends gpk, ok, and $(gsk_i)_{i\in[n]}$ to $\mathcal{A}$.

(ii) H_1 *query*. When $\mathcal{A}$ issues an H_2 query M, $\mathcal{B}$ retrieves a record (M, w, z, ν) for some w, z, and ν. If such a record is found, $\mathcal{B}$ replies with w. Otherwise, to respond to the query M to the random oracle H_1, $\mathcal{B}$ first flips a biased coin ν_M which is 0 with probability θ and is 1 with probability $1-\theta$ (the exact quantity of θ is determined later). Then $\mathcal{B}$ chooses a random integer $z_M \longleftarrow \mathbb{Z}_p$, computes w_M as

$$w_M = \begin{cases} g^{z_M} & (\nu_M = 0) \\ g^{\delta_2} g^{z_M} & (\nu_M = 1), \end{cases} \tag{D.2}$$

stores (M, w_M, z_M, ν_M) in the hash list for H_1, and returns w_M to $\mathcal{A}$.

(iii) H_2 *query*. When receiving a query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ to the random oracle H_2, if the response to this query is recorded in the hash list for H_2, $\mathcal{B}$ responds with the corresponding the record. Otherwise $\mathcal{B}$ chooses a random integer $c \longleftarrow \mathbb{Z}_p$, records $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c)$ to the hash list, and returns c to $\mathcal{A}$.

(iv) *Token query*. When $\mathcal{A}$ asks the token for a message M, $\mathcal{B}$ picks the record (M, w, z, ν) regarding the same query M to the random oracle H_1. Such a record will be found since we modify (if necessary) $\mathcal{A}$ to meet the restriction introduced at the beginning of the proof. If $\nu = 1$ $\mathcal{B}$ immediately outputs random $b \longleftarrow \{0,1\}$ and halts. Otherwise if $\nu = 0$ then $\mathcal{B}$ returns $t_M = (g^{\delta_1})^z$ to $\mathcal{A}$.

(v) *Challenge*. When $\mathcal{A}$ asks the challenge signature for (M^*, i_0, i_1), $\mathcal{B}$ picks the record (M^*, w, z, ν) regarding the query M^* to the random oracle H_1. If $\nu = 0$ $\mathcal{B}$ immediately outputs 0 and halts. Otherwise $\mathcal{B}$ chooses random $\alpha, \beta, \gamma, \eta \longleftarrow \mathbb{Z}_{\mathbb{p}}$, and computes $T_1^* = u^{\alpha}$, $T_2^* = v^{\beta}$, $T_3^* = h^{\alpha+\beta}$, $T_4^* = g_1^{\alpha} g_2^{\beta} A_{i_b} g^{\eta}$, $T_5^* = g^{\delta_3}$, $T_6^* = e(g,g)^{\tau} e(g^{\delta_1}, g^{\delta_3})^z e(g,g)^{-\eta}$. Finally $\mathcal{B}$ computes a zero-knowledge proof $(c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ as in Game 1. If there is a record $(M^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c)$ for some c, $\mathcal{B}$ halts with output $b' \longleftarrow \{0,1\}$. Otherwise it sends $\sigma^* = (T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ to $\mathcal{A}$.

(vi) *Guess*. Finally $\mathcal{A}$ outputs a guess b' and halts. $\mathcal{B}$ outputs 1 if $b = b'$ holds, otherwise outputs 0.

We define two bad events. The event E_{ch} denotes the event that $\mathcal{B}$ halts in the challenge phase due to the condition $\nu = 0$ for the record of M^*. The event E_{tk} denotes the event in which $\mathcal{B}$ halts because of the failure of responding to (one of) the token queries. Since B outputs 0 if E_{ch} or E_{tk} occurs, we have that

$$\begin{aligned} \mathrm{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda) &= \Big|\Pr\left[\mathcal{B}\left(g^{\delta_1}, g^{\delta_2}, g^{\delta_3}, e(g,g)^{\tau}\right) \longrightarrow 1 \mid \tau = \delta_1\delta_2\delta_3\right] \\ &\quad - \Pr\left[\mathcal{B}\left(g^{\delta_1}, g^{\delta_2}, g^{\delta_3}, e(g,g)^{\tau}\right) \longrightarrow 1 \mid \tau = \delta\right]\Big| \\ &= \Big|\Pr\left[b = b' \wedge \overline{E_{\mathrm{tk}}} \wedge \overline{E_{\mathrm{ch}}} \mid \tau = \delta_1\delta_2\delta_3\right] - \Pr\left[b = b' \wedge \overline{E_{\mathrm{tk}}} \wedge \overline{E_{\mathrm{ch}}} \mid \tau = \delta\right]\Big|. \end{aligned} \tag{D.3}$$

The event $\overline{E_{\mathrm{tk}}}$ occurs when $\nu_M = 0$ (with probability θ) for all M in the token query phase. Let q_{Td} denote the number of token queries issued by $\mathcal{A}$. If $\tau = \delta_1\delta_2\delta_3$, it holds that

$$\Pr\left[\overline{E_{\mathrm{tk}}} \mid \tau = \delta_1\delta_2\delta_3\right] = \theta^{q_{\mathsf{Td}}} \tag{D.4}$$

and the event $\overline{E_{\mathrm{ch}}}$ occurs when $\nu_{M^*} = 1$ (with probability $1-\theta$) for M^* in the challenge phase. Hence, if $\tau = \delta_1\delta_2\delta_3$, it holds that

$$\Pr\left[\overline{E_{\mathrm{ch}}} \mid \tau = \delta_1\delta_2\delta_3\right] = 1 - \theta. \tag{D.5}$$

Note that the events E_{tk} and E_{ch} are independent since each biased coin ν_M is flipped independently. Moreover, whether $\nu_M = 0$ or $\nu_M = 1$, the distributions of the values that $\mathcal{A}$ receives are identical, and hence $\mathcal{A}$'s view does not change. It means that the value of ν_M does not affect the behavior of $\mathcal{A}$. Therefore, the event E_{tk} and E_{ch} have no effect on the probability that $\mathcal{A}$ succeeds in guessing the challenge bit. That is, the event of $b = b'$, E_{tk}, E_{ch} are mutually independent.

From the above, we have the following equality.

$$\begin{aligned} &\Pr\left[b = b' \wedge \overline{E_{\mathrm{tk}}} \wedge \overline{E_{\mathrm{ch}}} \mid \tau = \delta_1\delta_2\delta_3\right] \\ &\quad = \Pr\left[b = b' \mid \tau = \delta_1\delta_2\delta_3\right] \cdot \Pr\left[\overline{E_{\mathrm{tk}}} \mid \tau = \delta_1\delta_2\delta_3\right] \\ &\qquad \cdot \Pr\left[\overline{E_{\mathrm{ch}}} \mid \tau = \delta_1\delta_2\delta_3\right] \\ &\quad = \theta^{q_{\mathsf{Td}}}(1-\theta)\Pr\left[b = b' \mid \tau = \delta_1\delta_2\delta_3\right] \\ &\quad = \theta^{q_{\mathsf{Td}}}(1-\theta)\Pr\left[S_1\right], \end{aligned} \tag{D.6}$$

where the last equality follows from the fact that given $\tau = \delta_1\delta_2\delta_3$ $\mathcal{B}$ correctly simulates Game 1 for $\mathcal{A}$. The detail follows. Firstly, the distribution of $\mathcal{B}$'s responses to the H_1

queries are identical to those of Game 1. This is because $w_M = g^{z_M}$ and $w_M = g^{\delta_2} g^{z_M}$ are both distributed uniformly over $\mathbb{G}$. Secondly, the responses to H_2 queries are also distributed identically to those of Game 1. Thirdly, the responses to token queries are also distributed identically. Notice that in Game 1 a token for a message M is computed as $H_1(M)^{\xi}$, while that in $\mathcal{B}$'s simulation is computed as $(g^{\delta_1})^{z_M}$ (if the simulation proceeds without an abort). Then we have that

$$H_1(M)^{\xi} = \left(g^{\log_g H_1(M)}\right)^{\xi} = \left(g^{\xi}\right)^{\log_g H_1(M)} = \left(g^{\delta_1}\right)^{z}, \quad \text{(D.7)}$$

where the last equality comes from the two facts that we set $y = g^{\delta_1}$ in $\mathcal{B}$'s simulation, which corresponds to $y = g^{\xi}$ in Game 1, and that we set $H_1(M) = g^{z_M}$ in the response of the H_1 query M. Lastly, the response to the challenge query is also distributed identically to that of Game 1. Notice that in Game 1 $T_5^* = g^{\rho}$ and $T_6^* = e(y, H_1(M))^{\rho} e(g,g)^{-\eta})$ for some $\rho, \eta \longleftarrow \mathbb{Z}_p$. Further, notice that in $\mathcal{B}$'s simulation, $T_5^* = g^{\delta_3}$ and $T_6^* = e(g,g)^{\tau} e(g^{\delta_1}, g^{\delta_3}) e(g,g)^{-\eta}$ (if the simulation proceeds without an abort). Therefore, g^{ρ} in Game 1 corresponds to g^{δ_3} in $\mathcal{B}$'s simulation. Hence if $\tau = \delta_1\delta_2\delta_3$, we have that

$$\begin{aligned} e(y, H_1(M))^{\rho} &= e\left(g^{\delta_1}, g^{\delta_2} g^{z}\right)^{\delta_3} \\ &= e(g,g)^{\delta_1\delta_2\delta_3} e\left(g^{\delta_1}, g^{\delta_3}\right)^{z} \\ &= e(g,g)^{\tau} e\left(g^{\delta_1}, g^{\delta_3}\right)^{z}. \end{aligned} \quad \text{(D.8)}$$

Therefore, we can conclude that given $\tau = \delta_1\delta_2\delta_3$ $\mathcal{B}$ correctly simulates Game 1.

Similarly, if $\tau = \delta$, namely $\mathcal{B}$ receives a random tuple, we have that

$$\begin{aligned} &\Pr\left[b = b' \wedge \overline{E_{\mathrm{tk}}} \wedge \overline{E_{\mathrm{ch}}} \mid \tau = \delta\right] \\ &= \theta^{q_{\mathsf{Td}}}(1-\theta) \cdot \Pr\left[b = b' \mid \tau = \delta\right] \\ &= \theta^{q_{\mathsf{Td}}}(1-\theta) \cdot \Pr\left[S_2\right], \end{aligned} \quad \text{(D.9)}$$

where the last equality comes from the fact that given that $\tau = \delta$ $\mathcal{B}$ correctly simulates Game 2. The correctness of the responses to the H_1 queries, H_2 queries, and token queries follows from a similar argument to that of the case of $\tau = \delta_1\delta_2\delta_3$. The correctness of the response to the challenge query also follows from a similar argument to that of the same case but replacing τ with δ ensures that T_6^* is distributed uniformly and independently.

Finally we obtain

$$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda) = \theta^{q_{\mathsf{Td}}}(1-\theta)\left|\Pr\left[S_1\right] - \Pr\left[S_2\right]\right|, \quad \text{(D.10)}$$

and hence

$$\begin{aligned} \left|\Pr\left[S_1\right] - \Pr\left[S_2\right]\right| \le (q_{\mathsf{Td}} + 1) \cdot \left(1 + \frac{1}{q_{\mathsf{Td}}}\right)^{q_{\mathsf{Td}}} \\ \cdot \mathrm{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda) \end{aligned} \quad \text{(D.11)}$$

when substituting θ with $q_{\mathsf{Td}}/(q_{\mathsf{Td}} + 1)$, which minimizes $1/\theta^{q_{\mathsf{Td}}}(1-\theta)$. Since $(1 + 1/q_{\mathsf{Td}})^{q_{\mathsf{Td}}} \le \exp(1)$, the lemma follows. □

These three lemmas conclude the entire proof. This complete the proof of Theorem 12. □

D.2. Proof of Theorem 13

Proof. The proof proceeds with a sequence of games. Let $\mathcal{A}$ be an adversary against the admitter anonymity. We define the following games, which $\mathcal{A}$ plays. In the following we denote by S_i the event that in Game i $\mathcal{A}$ successfully guesses the bit picked by the challenger.

(i) *Game 0.* The initial game is identical to the game defined in the definition of admitter anonymity. In order to respond to hash queries, the challenger maintains a hash list, which contains tuples of the form $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c)$ for the hash function H_2 and similarly for H_1.

(ii) *Game 1.* In Game 1, we replace the zero-knowledge proof, included in the challenge, with a simulated proof. More concretely, when the adversary asks a challenge by sending (i_0, i_1, M^*), the challenger proceeds as follows. The challenger first flips a bit b and encrypts A_{i_b} (a part of gsk_{i_b}) to obtain a ciphertext $(T_1^*, \ldots, T_6^*)$ as in Game 0. Then the challenger generates random integers $c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^* \longleftarrow \mathbb{Z}_p^*$, computes

$$\begin{aligned} R_1^* &\longleftarrow u^{s_\alpha^*}\left(T_1^*\right)^{-c^*}, \\ R_2^* &\longleftarrow v^{s_\beta^*}\left(T_2^*\right)^{-c^*}, \\ R_3^* &\longleftarrow h^{s_\alpha^* + s_\beta^*}\left(T_3^*\right)^{-c^*}, \\ R_4^* &\longleftarrow e\left(T_4^*, g\right)^{s_x^*} e\left(g_1, w\right)^{-s_\alpha^*} e\left(g_1, g\right)^{-s_{\alpha x}^*} \\ &\quad \cdot e\left(g_2, w\right)^{-s_\beta^*} e\left(g_2, g\right)^{-s_{\beta x}^*} \cdot e\left(g, w\right)^{-s_\eta^*} e\left(g, g\right)^{-s_{\eta x}^*} \\ &\quad \cdot \left(\frac{e(g,g)}{e\left(T_4^*, w\right)}\right)^{-c^*}, \\ R_5^* &\longleftarrow g^{s_\rho^*}\left(T_5^*\right)^{-c^*}, \\ R_6^* &\longleftarrow e\left(y, H_1(M)\right)^{s_\rho^*} e\left(g,g\right)^{-s_\eta^*}\left(T_6^*\right)^{-c^*}, \\ R_7^* &\longleftarrow \left(T_1^*\right)^{s_x^*} u^{-s_{\alpha x}^*}, \\ R_8^* &\longleftarrow \left(T_2^*\right)^{s_x^*} v^{-s_{\beta x}^*}, \\ R_9^* &\longleftarrow \left(T_5^*\right)^{s_x^*} g^{-s_{\rho x}^*}, \\ R_{10}^* &\longleftarrow \left(T_6^*\right)^{s_x^*} e\left(y, H_1(M)\right)^{-s_{\rho x}^*} e\left(g,g\right)^{s_{\eta x}^*}, \end{aligned} \quad \text{(D.12)}$$

and adds the tuple $(M^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c^*)$ to the list for H_1. If the hash list for H_2 contains a tuple of the form $(M, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c)$ with arbitrary

c, the challenger outputs $\perp$ and halts. If this is not the case, the challenge

$$\left(T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*\right) \quad \text{(D.13)}$$

is returned to the adversary. This change only causes a negligible difference to the advantage of $\mathcal{A}$. See Lemma D.4 for the details.

(iii) *Game 2*. In this game we modify the linear encryption in the challenge to be "invalid." More precisely, to compute the challenge signature $(T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$, the challenger selects random integers $\alpha, \beta \longleftarrow \mathbb{Z}_p$ and $\tau \longleftarrow \mathbb{Z}_p \setminus \{\alpha + \beta\}$, and computes

$$\begin{aligned} T_1^* &= u^\alpha, \\ T_2^* &= v^\beta, \\ T_3^* &= h^\tau, \\ T_4^* &= \left(T_1^*\right)^{\xi_1} \left(T_2^*\right)^{\xi_2} \left(T_3^*\right)^{\xi_3} A_{i_b} g^\eta, \end{aligned} \quad \text{(D.14)}$$

where u, v, and h are the part of the group public key gpk, b is the bit flipped for the challenge, A_{i_b} is the part of the group signing key of the member i_b, and η is the random integer which is also used to compute T_6^*. Notice that the challenger uses the opening key ok (actually its components ξ_1, ξ_2, and ξ_3) to compute the challenge. All the other components of the challenge are generated as in Game 1. This modification also does not change $\mathcal{A}$'s winning probability nonnegligibly, provided that the DLIN assumption holds. See Lemma D.5 for the details.

(iv) *Game 3*. In this game we modify the responses to opening queries in such a way that the challenger responds with $\perp$ if a queried signature $(T_1, \ldots, T_6, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ satisfies the following two conditions:

$$\left(M^*, T_1, \ldots, T_6\right) = \left(M^*, T_1^*, \ldots, T_6^*\right), \quad \text{(D.15)}$$

that is, the components $T_1, \ldots, T_6$ in the query are reused from the challenge signature, and

$$\left(R_1', \ldots, R_{10}'\right) = \left(R_1^*, \ldots, R_{10}^*\right) \quad \text{(D.16)}$$

where $(R_1', \ldots, R_{10}')$ is the group elements reproduced in the verification process. This change does not affect $\mathcal{A}$' advantage nonnegligibly. See Lemma D.6 for the details.

(v) *Game 4*. We further introduce another rejection rule in the responses to opening queries. This game rejects a signature that contains a ciphertext whose linear encryption components (T_1, T_2, T_3) does not constitute a linear tuple. Specifically when T_1, T_2, and T_3 satisfy $T_1 = u^\alpha$, $T_2 = v^\beta$, $T_3 = h^\upsilon$, the challenger immediately rejects queries if $\alpha + \beta \neq \upsilon$, and all other queries are treated as before. This modification does not affect the behavior of $\mathcal{A}$, as the adversary can issue such an invalid query with a valid proof (that passes the verification) only with a negligible probability. See Lemma D.7 for details.

The advantage of $\mathcal{A}$ is $|\Pr[S_0] - 1/2|$, and from the triangle inequality we can bound the advantage as the following:

$$\begin{aligned} \left|\Pr\left[S_0\right] - \frac{1}{2}\right| \leq \sum_{i=0}^{3} \left|\Pr\left[S_i\right] - \Pr\left[S_{i+1}\right]\right| \\ + \left|\Pr\left[S_4\right] - \frac{1}{2}\right|. \end{aligned} \quad \text{(D.17)}$$

To complete the proof it remains to prove the following lemmas.

The difference between Game 0 and Game 1 is bounded by a quite similar argument to that of Lemma D.2, hence we omit a detailed proof.

Lemma D.4. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible.*

We then bound the differences in $\mathcal{A}$'s success probability of the remaining game-hops and prove that $\mathcal{A}$'s advantage is zero in the last game.

Lemma D.5. $|\Pr[S_1] - \Pr[S_2]|$ *is negligible, provided that the DLIN assumption holds.*

Proof (of Lemma D.5). We will describe an algorithm $\mathcal{B}$ of the DLIN problem to bound the absolute difference $|\Pr[S_1] - \Pr[S_2]|$. $\mathcal{B}$ receives a tuple $(u, v, h, u^\alpha, v^\beta, h^\tau)$, in which τ is either $\alpha + \beta$ or not, together with the description $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ of the bilinear groups. $\mathcal{B}$ sets up the scheme by choosing $\xi_1, \xi_2, \xi_3, \zeta, \gamma, x_i \longleftarrow \mathbb{Z}_p$ $(i \in [n])$, setting $g_1 = u^{\xi_1} h^{\xi_3}$, $g_2 = v^{\xi_2} h^{\xi_3}$, $y = g^\zeta$, $w = g^\gamma$, and $A_i = g^{1/(\gamma + x_i)}$ $(i \in [n])$, and letting $gpk = (p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w)$, $ak = \zeta$, $ok = (\xi_1, \xi_2, \xi_3, (x_i)_{i \in [n]})$, and $(gsk_i)_{i \in [n]} = (A_i, x_i)_{i \in [n]}$. Then $\mathcal{B}$ runs $\mathcal{A}$ with an input $(gpk, ak, (gsk_i)_{i \in [n]})$. Queries from $\mathcal{A}$ to the random oracles H_1 and H_2 are responded in the ordinary manner, that is, all fresh queries are responded with a random hash value and are recorded together with the hash value, while previously issued queries are responded in the same way as in the previous query. Opening queries from $\mathcal{A}$ are responded as specified in the scheme, that is, $\mathcal{B}$ first verifies the NIZK proof. If the proof passes the verification, $\mathcal{B}$ decrypts the "linear encryption" part (T_1, T_2, T_3, T_4) using ξ_1, ξ_2, and ξ_3. If the signature does not pass the verification, $\mathcal{B}$ returns $\perp$. When $\mathcal{A}$ requests a challenge regarding (i_0, i_1, M), $\mathcal{B}$ proceeds as follows: To compute the challenge

$(T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$, $\mathscr{B}$ flips a bit b, chooses random integers $\rho, \eta \longleftarrow \mathbb{Z}_p$, and sets

$$\begin{aligned} T_1^* &= u^\alpha, \\ T_2^* &= v^\beta, \\ T_3^* &= h^\tau, \\ T_4^* &= \left(T_1^*\right)^{\xi_1} \left(T_2^*\right)^{\xi_2} \left(T_3^*\right)^{\xi_3} A_{i_b} g^\eta, \\ T_5^* &= g^\rho, \\ T_6^* &= e\left(y, H_1(M)\right)^\rho e\left(g, g\right)^{-\eta}. \end{aligned} \tag{D.18}$$

The zero-knowledge proof $(c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ is computed as in Game 1. Then $\mathscr{B}$ sends the challenge signature computed as above to $\mathscr{A}$. After receiving the challenge, $\mathscr{A}$ further makes queries to the random oracles and opening queries, which are responded as before by $\mathscr{B}$. Finally $\mathscr{A}$ outputs the guess b'. $\mathscr{B}$ outputs 1 if $b' = b$, outputs 0 otherwise.

Observe that when $\mathscr{B}$ receives a tuple satisfying $\tau \neq \alpha + \beta$, the adversary's view is equivalent to that of Game 2. In contrast, when $\mathscr{B}$ receives a linear tuple, we can see that $\mathscr{A}$'s view is identical to that of Game 1, as the equation $(T_1^*)^{\xi_1}(T_2^*)^{\xi_2}(T_3^*)^{\xi_3} = (u^\alpha)^{\xi_1}(v^\beta)^{\xi_2}(h^{\alpha+\beta})^{\xi_3} = (u^{\xi_1}h^{\xi_3})^\alpha (v^{\xi_2}h^{\xi_3})^\beta = g_1^\alpha g_2^\beta$ holds. Finally, it holds that

$$\begin{aligned} &\left|\Pr\left[S_1\right] - \Pr\left[S_2\right]\right| = \left|\Pr\left[\mathscr{B}\left(u, v, h, u^\alpha, v^\beta, h^{\alpha+\beta}\right)\right]\right. \\ &\left. - \Pr\left[\mathscr{B}\left(u, v, h, u^\alpha, v^\beta, h^\tau\right)\right]\right| = \mathrm{Adv}_{\mathscr{B}}^{\mathrm{DLIN}}(\lambda). \end{aligned} \tag{D.19}$$

When the DLIN assumption holds $\mathrm{Adv}_{\mathscr{B}}^{\mathrm{DLIN}}(\lambda)$ is negligible, therefore $|\Pr[S_1] - \Pr[S_2]|$ is negligible. □

Lemma D.6. $|\Pr[S_2] - \Pr[S_3]| \leq n/p$.

Proof (of Lemma D.6). Since Game 3 differs from Game 2 only when a queried signature, when verified, produces the same $(R_1, \ldots, R_{10})$ as the $(R_1^*, \ldots, R_{10}^*)$ used in the challenge phase, we examine the mapping $\psi : (s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x}, c) \longmapsto (R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10})$, implicitly defined by the verification algorithm (notice that the mapping ψ implicitly depends on the group public key *and the signature to be verified*), and argue that it is injective with overwhelming probability. Since $\mathscr{A}$ can issue queries satisfying (D.15) and (D.16) only when the mapping ψ is not injective. Then, to complete the proof we bound the probability that ψ is not injective.

$$\begin{pmatrix} \log_g u & & & & & & & & & -\log_g T_1 \\ & \log_g v & & & & & & & & -\log_g T_2 \\ \log_g h & \log_g h & & & & & & & & -\log_g T_3 \\ -\gamma \log_g g_1 & -\gamma \log_g g_2 & & & -\gamma & \log_g T_4 & -\log_g g_1 & -\log_g g_2 & & 1 \quad -\left(1 - \gamma \log_g T_4\right) \\ & & 1 & & & & & & & -\log_g T_5 \\ & & \zeta \log_g H_1(M) & -1 & & & & & & -\log_{e(g,g)} T_6 \\ & & & & \log_g T_1 & -\log_g u & & & & \\ & & & & \log_g T_2 & & -\log_g v & & & \\ & & & & \log_g T_5 & & & -1 & & \\ & & & & \log_{e(g,g)} T_6 & & & -\zeta \log_g H_1(M) & 1 & \end{pmatrix} \tag{D.20}$$

Since the mapping ψ is a linear function, by calculating the determinant of a matrix of (D.20), we can see that ψ is not injective if and only if

$$\begin{aligned} &-\left(\log_g h\right) \cdot \left(\log_g u\right)^2 \cdot \left(\log_g v\right)^2 \cdot (\alpha + \beta - \tau) \\ &\cdot \left(\frac{1}{x_{i_b} + \gamma} - (\alpha + \beta - \tau) \cdot \xi_3 \cdot \log_g h\right) = 0. \end{aligned} \tag{D.21}$$

Since $\mathscr{A}$ can issue queries satisfying (D.15) and (D.16) only when the mapping ψ is not injective, the difference $|\Pr[S_2] - \Pr[S_3]|$ is bounded by the probability that the above equation holds. Actually, (D.21) holds with probability at most n/p. This is because if (D.21) holds, there is $i \in [n]$ satisfying

$$\begin{aligned} &-\left(\log_g h\right) \cdot \left(\log_g u\right)^2 \cdot \left(\log_g v\right)^2 \cdot (\alpha + \beta - \tau) \\ &\cdot \left(\frac{1}{x_i + \gamma} - (\alpha + \beta - \tau) \cdot \xi_3 \cdot \log_g h\right) = 0. \end{aligned} \tag{D.22}$$

For a fixed i the probability that (D.22) holds is $1/p$, which can be seen by fixing all the variables except ξ_3 and seeing that ξ_i is still distributed uniformly. Hence from the union bound the probability that (D.21) holds is at most n/p. □

Lemma D.7. $|\Pr[S_3] - \Pr[S_4]|$ *is negligible.*

Proof (of Lemma D.7). Game 4 differs from Game 3 when $\mathcal{A}$ issues an opening query which is not rejected in Game 3 but is rejected in Game 4. We thus bound the probability that $\mathcal{A}$ issues such a query. More precisely, the event we consider is that $\mathcal{A}$ issues a signature $(T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ as an opening query such that

(i) it is verified as valid by GVf,

[ii] $(M, T_1, \ldots, T_6, R'_1, \ldots, R'_{10}) \neq (M^*, T_1^*, \ldots, T_3^*, R_1^*, \ldots, R_{10}^*)$, in which $(R'_1, \ldots, R'_{10})$ is the group elements computed in GVf as in Box 3 and $(R_1^*, \ldots, R_{10}^*)$ are the group elements used for generating the challenge signature as in (D.12), and

[iii] (T_1, T_2, T_3) does not constitute a linear tuple.

If $\mathcal{A}$ issues such a query, there should be a query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ in H_2 (issued by $\mathcal{A}$ explicitly or issued during the verification process of an opening query) such that (T_1, T_2, T_3) does not constitute a linear tuple, and the hash value $H_2(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ coincides with the unique challenge c that is determined from the public information $(T_1, \ldots, T_6)$ and the commitment $(R_1, \ldots, R_{10})$. Hence for concluding the proof it is sufficient to bound the probability of this event. Notice that in this case any query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ to H_2 in question is different from $(M^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*)$, for which the hash value is set in the challenge phase. Hence in this case, the output of H_2 is chosen from $\mathbb{Z}_p$ uniformly. Thus the hash value coincides with the unique value with probability $1/p$. Therefore, by the union bound, the probability that there is an opening query which satisfies the above three conditions is at most $(q_{H_2} + q_{\text{open}})/p$, where q_{H_2} is the number of H_2 queries issued by $\mathcal{A}$, and q_{open} is the number of opening queries issued by $\mathcal{A}$. □

Lemma D.8. $|\Pr[S_4] - 1/2| = 0$.

Proof (of Lemma D.8). Here we prove that in Game 4 the value $(T_1^*)^{\xi_1}(T_2^*)^{\xi_2}(T_3^*)^{\xi_3}$ is uniformly random even when conditioned on $\mathcal{A}$'s view. To this end we examine the distribution of $\mathcal{A}$'s view related to the randomness ξ_1, ξ_2, and ξ_3 under the condition where all the other randomness involved in the game are fixed. $\mathcal{A}$ obtains information related to ξ_1, ξ_2, and ξ_3 from the part of the group public key g_1 and g_2 and the responses to the opening queries. As for the responses to the opening queries, any query whose T_1, T_2, and T_3 components do not constitute the linear tuple will be rejected, thus $\mathcal{A}$ gains no information on ξ_1, ξ_2, and ξ_3 from such responses. A query with a linear tuple also gives no information to $\mathcal{A}$. When $\mathcal{A}$ issues a signature $(T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$, the game computes a group element

$$T_1^{\xi_1} T_2^{\xi_2} T_3^{\xi_3} \tag{D.23}$$

(the rest of the calculation is done without referring to ξ_1, ξ_2, and ξ_3), which is what $\mathcal{A}$ learns from this query. It in fact does not increase the information $\mathcal{A}$ knows, because the above equation can be rewritten as

$$\begin{aligned} T_1^{\xi_1} T_2^{\xi_2} T_3^{\xi_3} &= (u^\alpha)^{\xi_1} \left(v^\beta\right)^{\xi_2} \left(h^{\alpha+\beta}\right)^{\xi_3} \\ &= \left(u^{\xi_1} h^{\xi_3}\right)^\alpha \left(v^{\xi_2} h^{\xi_3}\right)^\beta = (g_1)^\alpha (g_2)^\beta, \end{aligned} \tag{D.24}$$

when we write $T_1 = u^\alpha$, $T_2 = v^\beta$, and $T_3 = h^{\alpha+\beta}$. The last formula of the equation shows that responses of this type give no information to $\mathcal{A}$, since all the values that appear in the formula are already known to $\mathcal{A}$.

The above discussion shows that the responses to the opening queries do not leak any information of ξ_1, ξ_2, and ξ_3 beyond the equations of

$$\log_g g_1 = \xi_1 \log_g u + \xi_3 \log_g h \tag{D.25}$$

and

$$\log_g g_2 = \xi_2 \log_g v + \xi_3 \log_g h \tag{D.26}$$

to $\mathcal{A}$. Finally we show that the value $(T_1^*)^{\xi_1}(T_2^*)^{\xi_2}(T_3^*)^{\xi_3}$ is uniformly distributed conditioned on the elements g_1 and g_2. This can be seen by considering the following equation

$$\begin{aligned} &\begin{pmatrix} \log_g g_1 \\ \log_g g_2 \\ \log_g (T_1^*)^{\xi_1} (T_2^*)^{\xi_2} (T_3^*)^{\xi_3} \end{pmatrix} \\ &= \begin{pmatrix} \log_g u & & \log_g h \\ & \log_g v & \log_g h \\ \alpha \log_g u & \beta \log_g v & \tau \log_g h \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix}. \end{aligned} \tag{D.27}$$

Since the matrix in the right-hand side has the determinant

$$\left(\log_g u\right) \cdot \left(\log_g v\right) \cdot \left(\log_g h\right) \cdot (\tau - \alpha - \beta) \neq 0, \tag{D.28}$$

the value $(T_1^*)^{\xi_1}(T_2^*)^{\xi_2}(T_3^*)^{\xi_3}$ is distributed uniformly and independently of g_1 and g_2. This shows that the challenge signature is independent of A_{i_b} and hence of the challenge bit b. This completes the proof of Lemma D.8. □

From the above, the proof of Theorem 13 is completed. □

D.3. Proof of Theorem 14

Proof. Suppose $\mathcal{A}$ is an adversary that attacks the traceability of the GS-MDO scheme. We assume that if the adversary $\mathcal{A}$ outputs a message-signature pair $(M, (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x}))$, $\mathcal{A}$ issued the H_2 query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ in which R_1, …, R_{10} are the reproduced R-values in the verification of the output signature. We also assume that before issuing H_2 query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ $\mathcal{A}$ issues the H_1 query M. We do not lose generality by these restrictions, since given an adversary $\mathcal{A}$ which not necessarily meets this restrictions, we can modify $\mathcal{A}$ to one that meets the restrictions. In addition,

we assume that $\mathcal{A}$ issues q_{H_1} H_1 queries, q_{H_2} H_2 queries, and q_{sign} signing queries. Let n be the number of the group members.

We consider the following sequence of games.

(i) *Game 0*. The initial game is identical to the traceability game.

Let S_i be the event that in Game i $\mathcal{A}$ satisfies the winning condition. We have that

$$\begin{aligned}\Pr[S_0] &= \Pr[S_0] - \Pr[S_1] + \Pr[S_1] - \Pr[S_2] \\ &\quad + \Pr[S_2] \\ &\leq \left|\Pr[S_0] - \Pr[S_1]\right| + \left|\Pr[S_1] - \Pr[S_2]\right| \\ &\quad + \Pr[S_2].\end{aligned} \tag{D.29}$$

We then bound each term.

The difference between Game 0 and Game 1 is bounded by a similar argument to Lemma D.2. Hence we omit the proof.

Lemma D.9. $|\Pr[S_0] - \Pr[S_1]|$ *is negligible.*

We then bound the difference between Game 1 and Game 2.

Lemma D.10. $|\Pr[S_1] - \Pr[S_2]|$ *is negligible if the n-SDH assumption holds.*

Proof. Let F be the event that $\mathcal{A}$ outputs a valid forgery which satisfies the clause *(i)* in the traceability game. By the difference lemma [67] we have that $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F]$. We bound this $\Pr[F]$ by constructing an n-SDH adversary $\mathcal{B}$.

The construction proceeds with the forking lemma [66]. We first construct an instance generator: It first generates a description of bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g') \longleftarrow \mathcal{G}(1^\lambda)$; then chooses a random integer $\gamma \longleftarrow \mathbb{Z}_p$ and outputs $(g', g'^{\gamma}, \ldots, g'^{\gamma^n})$. Given the adversary $\mathcal{A}$, we construct another algorithm $\mathcal{A}'$, which takes as an input an n-SDH instance $(g', g'^{\gamma}, g'^{\gamma^2} \ldots, g'^{\gamma^n})$ and a sequence of random exponents $(c_1, \ldots, c_{q_{H_2}})$ and outputs a tuple (j, σ) where $j \in \{0, \ldots, q_{H_2}\}$. Here we assume that if $\mathcal{A}$ outputs a forgery (M^*, σ^*) then $\mathcal{A}$ issues a H_2 query $(M, T_1, \ldots, T_6, R_1', \ldots, R_{10}')$ that will be issued during the verification of the forged signature σ^*. The construction of $\mathcal{A}'$ is as follows.

(i) *Setup*. $\mathcal{A}'$ takes as inputs an n-SDH instance $(g', g'^{\gamma}, g'^{\gamma^2} \ldots, g'^{\gamma^n})$ and a sequence of random exponents $(c_1, \ldots, c_{q_{H_2}})$, and sets up n SDH pairs (x_i, A_i) where $x_i \longleftarrow \mathbb{Z}_p$ and $A_i = g^{1/(x_i+\gamma)}$ together with group elements g and $w = g^{\gamma}$ (in the same way as done in the proof of Lemma 9 of [27]). $\mathcal{A}'$ chooses random group elements $u, v, h \longleftarrow \mathbb{G} \setminus \{1\}$ and random exponents $\xi_1, \xi_2, \xi_3, \zeta \longleftarrow \mathbb{Z}_p$, and sets $g_1 \longleftarrow u^{\xi_1} h^{\xi_3}$, $g_2 \longleftarrow v^{\xi_2} h^{\xi_3}$, and $y \longleftarrow g^{\zeta}$. It then sets a counter $J \longleftarrow 0$. It generates a random tape rnd for $\mathcal{A}$. It finally sets $gpk \longleftarrow (p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w)$, $ak \longleftarrow \zeta$, and $ok \longleftarrow (\xi_1, \xi_2, \xi_3, (e(A_i, g))_{i \in [n]})$, and runs the adversary $\mathcal{A}(gpk, ok, ak; \mathsf{rnd})$.

(ii) H_1 *query*. When the adversary $\mathcal{A}$ issues an H_1 query M, $\mathcal{A}'$ searches for a recorded tuple (M, z) and returns z if one is found. If not, $\mathcal{A}'$ generates a random $z \longleftarrow \mathbb{G}$, records (M, z), and returns z to $\mathcal{A}$.

(iii) H_2 *query*. When the adversary $\mathcal{A}$ issues an H_2 query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ $\mathcal{A}'$ searches for a recorded tuple $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, j)$ for some c and j, and returns c if such a tuple is found. If not, $\mathcal{A}'$ increments the counter $J \longleftarrow J + 1$, records $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c_J, J)$, and returns c_J to $\mathcal{A}'$

(iv) *Key Revealing Query*. When the adversary $\mathcal{A}$ issues a key revealing query for the user i, $\mathcal{A}'$ returns (A_i, x_i) to $\mathcal{A}$.

(v) *Signing Query*. When the adversary $\mathcal{A}$ issues a signing query (i, M), $\mathcal{A}'$ generates a signature $\sigma = (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ as in (14), and confirms whether $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, j)$ for some c and some j is recorded. If recorded, $\mathcal{A}'$ outputs $(0, \perp)$ and halts. If not, $\mathcal{A}'$ records $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, \perp)$, and returns σ to $\mathcal{A}$.

(vi) *Output*. When the adversary $\mathcal{A}$ outputs a forgery (M^*, σ^*), let σ^* be $(T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ and $R_1^*, \ldots, R_{10}^*$ be the values reproduced in the verification of σ^*. Then $\mathcal{A}'$ decrypts $(T_1^*, \ldots, T_6^*)$ with ξ_1, ξ_2, ξ_3, and ζ by computing

$$U \longleftarrow e\left(\frac{T_4^*}{T_1^{*\xi_1} T_2^{*\xi_2} T_3^{*\xi_3}}, g\right) \cdot \frac{T_6^*}{e\left(H_1(M^*)^{\zeta}, T_5^*\right)} \tag{D.30}$$

and searches for $j \in [q_{H_2}]$ such that $(T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c, j)$ is recorded. Finally, if

(a) $\mathsf{GVf}(gpk, M^*, \sigma^*) = 1$,
(b) $U \notin \{e(g, A_1), \ldots, e(g, A_n)\}$, and
(c) such $j \in [q_{H_2}]$ is found,

$\mathcal{A}'$ outputs $(j, (H_1(M^*), \sigma^*))$. Otherwise it outputs $(0, \perp)$. The first condition *(a)* is well-defined, since we assume that the H_2 query will be issued during the verification of σ^*, and then to verify the condition *(a)* we do not need any extra H_2 query.

Now we argue that whenever F happens, $\mathcal{A}'$ outputs $(j, (z, \sigma))$ for some $j \neq 0$. We can see that $\mathcal{A}'$ perfectly simulates Game 2 for $\mathcal{A}$ (and Game 3). Let us assume F happens in this simulation. Then in the simulation by $\mathcal{A}'$, the first two conditions *(a)* and *(b)* are satisfied. To claim that the condition *(c)* is satisfied, we argue that the tuple $(M^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c, \perp)$ is not recorded when responding to the signing queries. It follows from the fact that $(T_1^*, \ldots, T_6^*)$ encrypts none of $e(A_i, g)$'s, while any $(T_1, \ldots, T_6)$ which is recorded when responding to the signing queries encrypts one of $e(A_i, g)$'s.

Hence whenever F happens, $\mathcal{A}'$ outputs $(j,(z,\sigma))$ with some $j \neq 0$.

We then apply the forking lemma [66] (See Appendix A for the statement) to $\mathcal{A}'$ and obtain the forking algorithm $\mathcal{F}_{\mathcal{A}'}$. Here, the number q_{H_2} of H_2 queries is assigned to q in the lemma, while $\mathbb{Z}_p$ is assigned to the set H. All the randomness but that for determining the responses to H_2 queries is assigned to rnd. The forking algorithm outputs $(1,(z^*,\sigma^*),(z^{**},\sigma^{**}))$ with probability frk such that $\Pr[F] \leq (q_{H_2}/p) + \sqrt{q_{H_2} \cdot \mathsf{frk}}$. Notice that $\Pr[F]$ is equal to acc in the forking lemma.

We then construct an algorithm $\mathcal{B}$ which solves the n-SDH problem whenever $\mathcal{F}_{\mathcal{A}'}$ outputs $(1,(z^*,\sigma^*),(z^{**},\sigma^{**}))$. The construction of $\mathcal{B}$ is as follows. Given n-SDH instance $(g', g'^{\gamma}, \ldots, g'^{\gamma^n})$, $\mathcal{B}$ runs the forking algorithm $\mathcal{F}_{\mathcal{A}'}(g', g'^{\gamma}, \ldots, g'^{\gamma^n})$. If $\mathcal{F}_{\mathcal{A}'}$ outputs a tuple $(0,\perp,\perp)$, $\mathcal{B}$ outputs $\perp$ and halts. If $\mathcal{F}_{\mathcal{A}'}$ outputs a tuple $(1,(z^*,\sigma^*),(z^{**},\sigma^{**}))$, where $\sigma^* = (T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^* s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ and $\sigma^{**} = (T_1^{**}, \ldots, T_6^{**}, c^{**}, s_\alpha^{**}, s_\beta^{**}, s_\rho^{**}, s_\eta^{**}, s_x^{**} s_{\alpha x}^{**}, s_{\beta x}^{**}, s_{\rho x}^{**}, s_{\eta x}^{**})$ it computes $\alpha^* = (s_\alpha^{**} - s_\alpha^*)/(c^{**} - c^*)$, and computes β^*, ρ^*, η^*, x^*, δ_1^*, δ_2^*, δ_3^*, and δ_4^* in a similar way, where δ_1^*, δ_2^*, δ_3^*, and δ_4^* are supposedly equal to $\alpha^* x^*$, $\beta^* x^*$, $\rho^* x^*$, and $\eta^* x^*$, respectively. Then $\mathcal{B}$ computes

$$A^* \longleftarrow \frac{T_4^*}{{g_1}^{\alpha^*} {g_2}^{\beta^*} g^{\eta^*}} \tag{D.31}$$

and obtains an SDH pair (x^*, A^*). Finally, $\mathcal{B}$ obtains a solution for the SDH instance from the SDH pair (x^*, A^*) (again, in the same way as done in the proof of Lemma 9 of [27]) and outputs this solution.

We show that whenever $\mathcal{F}_{A'}$ outputs $(1,\sigma^*,\sigma^{**})$, $\mathcal{B}$ obtains a new SDH pair and thus solves the SDH problem.

Firstly, we argue that $c^* \neq c^{**}$. This is due to the construction of $\mathcal{A}'$ and $\mathcal{F}_{\mathcal{A}'}$. Since we are assuming that $\mathcal{F}_{\mathcal{A}'}$ outputs $(1,(z^*,\sigma^*),(z^{**},\sigma^{**}))$, the first run of $\mathcal{A}'$ outputs $(j^*,(z^*,\sigma^*))$ where $j^* \in [q_{H_2}]$, and the second run of $\mathcal{A}'$ outputs $(j^{**},(z^{**},\sigma^{**}))$ where $j^{**} \in [q_{H_2}]$. Therefore, c^* is the j^*-th H_2 query in the first run, while c^{**} is the j^{**}-th H_2 query in the second run. Furthermore, since $\mathcal{F}_{\mathcal{A}'}$ outputs $(1,(z^*,\sigma^*),(z^{**},\sigma^{**}))$, we have that $j^* = j^{**}$ and $c^* \neq c^{**}$.

Secondly, to establish that $\mathcal{B}$ successfully computes α^* β^*, ρ^*, η^*, x^*, δ_1^*, δ_2^*, δ_3^*, and δ_4^*, we show that if the forking algorithm outputs $(1,(z^*,\sigma^*),(z^{**},\sigma^{**}))$, the equation

$$\begin{aligned}&(z^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*)\\ &\quad = (z^{**}, T_1^{**}, \ldots, T_6^{**}, R_1^{**}, \ldots, R_{10}^{**}),\end{aligned} \tag{D.32}$$

holds, where $R_1^*, \ldots, R_{10}^*$, and $R_1^{**}, \ldots R_{10}^{**}$ are the reproduced values in the verification of σ^* and σ^{**}, respectively. The equation holds because the random tapes of both runs are equal, the first $j^* - 1$ (and $j^{**} - 1$, respectively) responses of the H_2 queries are equal, H_1-queries M^* are issued before j^*-th (and j^{**}-th) H_2-queries are issued, where j^* and j^{**} are the outputs of the first and second runs of $\mathcal{A}'$ in the forking algorithm $\mathcal{F}_{\mathcal{A}}$. Then (D.32) holds because $(T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*)$ is the j^*-th H_2 query in the first run, $(T_1^{**}, \ldots, T_6^{**}, R_1^{**}, \ldots, R_{10}^*)$ is the j^{**}-th H_2 query in the second run, and H_1 query M^* (and M^{**}) is issued before the j^*-th (and j^{**}-th) H_2 query.

Thirdly, we show that the extracted pair (x^*, A^*) (in (D.31)) constitutes a new SDH pair. To show this, we claim that

$$\begin{aligned}e\left(\frac{T_4^*}{{g_1}^{\alpha^*} {g_2}^{\beta^*} g^{\eta^*}}, g\right) &= e\left(\frac{T_4^*}{{T_1^*}^{\xi_1} {T_2^*}^{\xi_2} {T_3^*}^{\xi_3}}, g\right)\\ &\quad \cdot \frac{T_6^*}{e\left((z^*)^{\zeta}, T_5^*\right)},\end{aligned} \tag{D.33}$$

in which the α, β, and η in the left-hand side are what $\mathcal{B}$ extracts, and the right-hand side is what $\mathcal{A}'$ computes when confirming the winning condition of $\mathcal{A}$. This equation can be verified from the fact that there is the witness that satisfies (10a), (10b), (10c), (10d), (10e), (10f), (10g), (10h), (10i), and (10j). Now we are considering the event that $\mathcal{F}_{\mathcal{A}'}$ is successful in forking, and thus $\mathcal{A}$ is successful in satisfying the clause *(i)* in the traceability game. Therefore, the right-hand side is not equal to any of $e(A_i, g)$ $(i \in [n])$. This assures that the extracted SDH pair (x^*, A^*) is a new pair.

Finally we have that whenever $\mathcal{B}$ is successful in forking, we can successfully obtain a solution to the SDH instance. This implies that frk is negligible, and thus $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F] \leq (q_{H_2}/p) + \sqrt{q_{H_2} \cdot \mathsf{frk}}$ is negligible. □

Lemma D.11. $\Pr[S_2]$ *is negligible if the* $(n-1)$*-SDH assumption holds.*

Proof. Using the adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$ that solves the n-SDH problem. The proof again proceeds with the forking lemma [66] (Also see Appendix A).

Given the adversary $\mathcal{A}$ we construct another algorithm $\mathcal{A}'$ which takes as an input an n-SDH instance $(g', g'^{\gamma}, \ldots, g'^{\gamma^{n-1}})$ and a sequence of random exponents $(c_1, \ldots, c_{q_{H_2}})$ and outputs a tuple (j,σ) such that $j \in \{0, \ldots, q_{H_2}\}$. The construction of $\mathcal{A}'$ is as follows.

(i) *Setup*. $\mathcal{A}'$ takes as an input an n-SDH instance $(g', g'^{\gamma}, \ldots, g'^{\gamma^{n-1}})$ and a sequence of random exponents $(c_1, \ldots, c_{q_{H_2}})$. Then $\mathcal{A}'$ chooses a random integer $i^* \longleftarrow [n]$ and sets up $n-1$ SDH pairs (x_i, A_i) where $i \in [n] \setminus \{i^*\}$ together with group elements g, $w = g^{\gamma}$, as in the proof of Lemma D.10. The algorithm then chooses a random group element $A_{i^*} \longleftarrow \mathbb{G}$. It then chooses $u, v, h \longleftarrow \mathbb{G} \setminus \{1\}$, chooses $\xi_1, \xi_2, \xi_3, \zeta \longleftarrow \mathbb{Z}_p$ and sets $g_1 \longleftarrow u^{\xi_1} h^{\xi_3}$, $g_2 \longleftarrow v^{\xi_2} h^{\xi_3}$, and $y \longleftarrow g^{\zeta}$. The algorithm sets up a counter $J \longleftarrow 0$. It finally sets $gpk = (p, \mathbb{G}, \mathbb{G}_T, e, g, u, v, h, g_1, g_2, y, w)$, $ok \longleftarrow (\xi_1, \xi_2, \xi_3, (e(A_i, g))_{i \in [n]})$, $ak \longleftarrow \zeta$, and $gsk_i \longleftarrow (x_i, A_i)$ for all $i \in [n] \setminus \{i^*\}$, chooses a random tape rnd for $\mathcal{A}$, and runs $\mathcal{A}(gpk, ok, ak; \mathsf{rnd})$.

(ii) H_1 *query*. When the adversary $\mathcal{A}$ issues an H_1 query M, $\mathcal{A}'$ searches for a recorded tuple (M, z) and

returns z if one is found. Otherwise, $\mathcal{A}'$ generates a random $z \longleftarrow \mathbb{G}$, records (M, z), and returns z to $\mathcal{A}$.

(iii) H_2 *query*. When the adversary $\mathcal{A}$ issues an H_2 query $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10})$ $\mathcal{A}'$ searches for a recorded tuple $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, j)$ for some c and j and returns c if one is found. If not, $\mathcal{A}'$ increments the counter $J \longleftarrow J + 1$, records $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c_J, J)$, and returns c_J.

(iv) *Key Revealing Query*. When the adversary $\mathcal{A}$ issues a key revealing query for the user i, if $i \neq i^*$, $\mathcal{A}'$ returns (A_i, x_i) to $\mathcal{A}$. If $i = i^*$, $\mathcal{A}'$ outputs $\perp$ and halts.

(v) *Signing Query*. When the adversary $\mathcal{A}$ issues a signing query (i, M), $\mathcal{A}'$ generates a signature $\sigma = (T_1, \ldots, T_6, c, s_\alpha, s_\beta, s_\rho, s_\eta, s_x, s_{\alpha x}, s_{\beta x}, s_{\rho x}, s_{\eta x})$ as in (14), and confirms whether $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, j)$ for some c and j is recorded. If recorded, $\mathcal{A}'$ outputs $(0, \perp)$ and halts. Otherwise, $\mathcal{A}'$ records $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, \perp)$, and returns σ to $\mathcal{A}$.

(vi) *Output*. When the adversary $\mathcal{A}$ outputs a forgery (M^*, σ^*), let σ^* be $(T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ and R_1^*, …, R_{10}^* be the R-values reproduced in the verification of σ^*. Then $\mathcal{A}'$ decrypts $(T_1^*, \ldots, T_6^*)$ with ξ_1, ξ_2, ξ_3, and ζ by computing

$$U \longleftarrow e\left(\frac{T_4^*}{T_1^{*\xi_1} T_2^{*\xi_2} T_3^{*\xi_3}}, g\right) \cdot \frac{T_6^*}{e\left(H_1(M^*)^\zeta, T_5^*\right)}, \quad \text{(D.34)}$$

and searches for $j \in [q_{H_2}]$ such that $(T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c, j)$ is recorded. Finally, if

(a) the clause *(ii)* of the winning condition is satisfied,

(b) $U = e(A_{i^*}, g)$, and

(c) such $j \in [q_{H_2}]$ is found,

$\mathcal{A}'$ outputs $(j, (H_1(M^*), \sigma^*))$. Otherwise it outputs $(0, \perp)$.

Now we argue that with probability $\Pr[S_2]/n$, $\mathcal{A}'$ outputs (j, σ) for some $j \in [q_{H_2}]$ and σ. Let (j, σ) denote the output of $\mathcal{A}'$. We want to show that

$$\begin{aligned} \Pr[j \neq 0] &= \Pr[S_2 \wedge \text{ the forgery is opened to } i^*] \\ &= \frac{\Pr[S_2]}{n}, \end{aligned} \quad \text{(D.35)}$$

where the last equality follows from the fact that i^* is independent of $\mathcal{A}$'s view. The simulation by $\mathcal{A}'$ is perfect except for the abort in the simulation of the responses to key revealing queries. Therefore, to show the first equality, it is sufficient to show that if the conditions *(a)* and *(b)* hold, the condition *(c)* also holds. Let us assume the conditions *(a)* and *(b)* hold. Then we can assume (i^*, M^*) is not queried as a signing query. Hence when responding to the signing queries, only tuples of the form $(M, T_1, \ldots, T_6, R_1, \ldots, R_{10}, c, \perp)$ such that $M \neq M^*$ or $(T_1, \ldots, T_6)$ does not encrypt $e(A_{i^*}, g)$ are recorded. Therefore, when the H_2 query of the form $(M^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*)$ where T_1^*, …, T_6^* are taken from the forgery σ^*, and R_1^*, …, R_{10}^* are the reproduced values in the verification of σ^*, the tuple of the form $(M^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*, c, j)$ is not recorded for none of c and j. Thus when responding to such an H_2 query a new tuple of such form is recorded. This shows that when the conditions *(a)* and *(b)* hold the condition *(c)* also holds.

We then apply the forking lemma to $\mathcal{A}'$. Here, the number q_{H_2} of H_2 queries is assigned to q, the set $\mathbb{Z}_p$ is assigned to H, and all the randomness used by $\mathcal{A}'$ except for that used to determine the responses to H_2 queries is set to rnd. We can obtain the forking algorithm $\mathcal{F}_{\mathcal{A}'}$ which outputs some $(1, (z^*, \sigma^*), (z^{**}, \sigma^{**}))$ with probability frk such that $\Pr[j \neq 0] \leq (q_{H_2}/p) + \sqrt{q \cdot \mathsf{frk}}$. Notice that $\Pr[F]$ is equal to acc in the forking lemma.

We then construct an algorithm $\mathcal{B}$ which solves the n-SDH problem using $\mathcal{F}_{\mathcal{A}'}$. The construction of $\mathcal{B}$ is as follows. Given an n-SDH instance $(g', g'^{\gamma}, \ldots, g'^{\gamma^n})$, $\mathcal{B}$ runs the forking algorithm $\mathcal{F}_{\mathcal{A}'}(g', g'^{\gamma}, \ldots, g'^{\gamma^n})$. If $\mathcal{F}_{\mathcal{A}'}$ outputs a tuple $(0, \perp, \perp)$, $\mathcal{B}$ outputs $\perp$ and halts. If $\mathcal{F}_{\mathcal{A}'}$ outputs $(1, (z^*, \sigma^*), (z^{**}, \sigma^{**}))$, where $\sigma^* = (T_1^*, \ldots, T_6^*, c^*, s_\alpha^*, s_\beta^*, s_\rho^*, s_\eta^*, s_x^*, s_{\alpha x}^*, s_{\beta x}^*, s_{\rho x}^*, s_{\eta x}^*)$ and $\sigma^{**} = (T_1^{**}, \ldots, T_6^{**}, c^{**}, s_\alpha^{**}, s_\beta^{**}, s_\rho^{**}, s_\eta^{**}, s_x^{**}, s_{\alpha x}^{**}, s_{\beta x}^{**}, s_{\rho x}^{**}, s_{\eta x}^{**})$, it computes $\alpha^* = (s_\alpha^{**} - s_\alpha^*)/(c^{**} - c^*)$ and computes β^*, ρ^*, η^*, x^*, δ_1^*, δ_2^*, δ_3^*, and δ_4^* in a similar manner, where δ_1^*, δ_2^*, δ_3^*, and δ_4^* are supposedly equal to $\alpha^* x^*$, $\beta^* x^*$, $\rho^* x^*$, and $\eta^* x^*$, respectively. Then $\mathcal{B}$ computes

$$A^* \longleftarrow \frac{T_4^*}{{g_1}^{\alpha^*} {g_2}^{\beta^*} g^{\eta^*}} \quad \text{(D.36)}$$

and obtains a new SDH pair (x^*, A^*). Finally $\mathcal{B}$ obtains a solution to the SDH instance from the new SDH pair and outputs this solution, as in the proof of Lemma D.10.

Firstly we argue that when the forking algorithm $\mathcal{F}_{\mathcal{A}'}$ outputs $(1, (z^*, \sigma^*), (z^{**}, \sigma^{**}))$, it holds that $c^* \neq c^{**}$. This is due to the constructions of $\mathcal{A}'$ and $\mathcal{F}_{\mathcal{A}'}$. Since $\mathcal{F}_{\mathcal{A}'}$ outputs $(1, (z^*, \sigma^*), (z^{**}, \sigma^{**}))$, the first run of $\mathcal{A}'$ outputs $(j^*, (z^*, \sigma^*)$ for some $j^* \in [q_{H_2}]$, and the second run of $\mathcal{A}'$ outputs $(j^{**}, (z^{**}, \sigma^{**}))$ for some $j^{**} \in [q_{H_2}]$. Therefore, due to the construction of $\mathcal{A}'$, c^* is the j^*-th H_2 query in the first run, and c^{**} is the j^{**}-th H_2 query in the second run. Furthermore, due to the construction of $\mathcal{F}_{\mathcal{A}'}$, we have that $j^* = j^{**}$ and $c^* \neq c^{**}$.

Secondly we claim that when the forking algorithm outputs $(1, (z^*, \sigma^*), (z^{**}, \sigma^*))$, the equation

$$\begin{aligned} &(z^*, T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*) \\ &\quad = (z^{**}, T_1^{**}, \ldots, T_6^{**}, R_1^{**}, \ldots, R_{10}^{**}) \end{aligned} \quad \text{(D.37)}$$

holds, where $R_1^*, \ldots, R_{10}^*$ and $R_1^{**}, \ldots, R_{10}^{**}$ are the reproduced values in the verification of σ^* and σ^{**}, respectively. This equation holds because the random tapes of both runs are equal, the responses to the first $j^* - 1$ (and $j^{**} - 1$) H_2 queries

are equal, and the H_1 queries M^* (of both runs) are issued before the j^*-th (and j^{**}-th) H_2 queries. Then (D.37) holds because $(T_1^*, \ldots, T_6^*, R_1^*, \ldots, R_{10}^*)$ is the j^*-th H_2 query, and $(z^{**}, T_1^{**}, \ldots, T_6^{**}, R_1^{**}, \ldots, R_{10}^{**})$ is the j^{**}-th H_2 query.

Thirdly we claim that the pair $\mathscr{B}$ obtained (as in (D.36)) constitutes a new SDH pair. To show this we claim that

$$e\left(\frac{T_4^*}{{g_1}^{\alpha^*} {g_2}^{\beta^*} g^{\eta}}, g\right) = e\left(\frac{T_4^*}{{T_1^*}^{\xi_1} {T_2^*}^{\xi_2} {T_3^*}^{\xi_3}}, g\right) \cdot \frac{T_6^*}{e\left(z^{*\zeta}, T_5^*\right)}, \quad \text{(D.38)}$$

where the left-hand side is what $\mathscr{B}$ extracts in (D.36), while the right-hand side is what $\mathscr{A}'$ computes for verifying the winning condition of $\mathscr{A}$. This equation is verified by a simple calculation as in the proof of Lemma D.10. This equation ensures that extracted A^* is equal to A_{i^*}, because the right-hand side is equal to $e(A_{i^*}, g)$, due to the constructions of $\mathscr{A}$. Note that the corresponding exponent of A_{i^*} is unknown to $\mathscr{B}$, hence the extracted SDH pair (x^*, A^*) is a new SDH pair.

Eventually, we have that whenever the forking is successful, $\mathscr{B}$ successfully obtains a new SDH pair. This implies that frk is negligible, and thus $\Pr[S_2] = n \cdot \Pr[j \neq 0] \leq (n \cdot q_{H_2}/p) + n\sqrt{q_{H_2} \cdot \mathsf{frk}}$ is negligible. □

From the above lemmas, the proof of Theorem 14 is completed. □

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

Preliminary versions of this paper are presented at the 5th International Conference on Pairing-Based Cryptography (Pairing 2012) and at the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2013). The present address of Kazuma Ohara is NEC Corporation, 1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666, Japan.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to Hovav Shacham for pointing out the possibility of using random oracles to achieve unbounded security. The authors would like to thank Shin-Akarui-Angou-Benkyou-Kai for valuable discussions and comments. A part of this work was supported by JSPS KAKENHI Grants numbers 18K18055 and 19H04107, Japan, and JST CREST Grant number JPMJCR19F6, Japan.

References

[1] D. Chaum and E. van Heyst, "Group signatures," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, Advances in Cryptology – EUROCRYPT'91*, D. W. Davies, Ed., vol. 547 of *Lecture Notes in Computer Science*, pp. 257–265, Springer, Berlin, Germany, April 1991.

[2] E. Ghadafi, "Efficient distributed tag-based encryption and its application to group signatures with efficient distributed traceability," in *Proceedings of the 3rd International Conference on Cryptology and Information Security in Latin America, Florianópolis, Brazil, Progress in Cryptology - LATINCRYPT '14*, D. F. Aranha and A. Menezes, Eds., vol. 8895 of *Lecture Notes in Computer Science*, pp. 327–347, Springer International Publishing, 2015.

[3] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, Advances in Cryptology – EUROCRYPT '03*, E. Biham, Ed., vol. 2656 of *Lecture Notes in Computer Science*, pp. 614–629, Springer, Berlin, Germany, May 2003.

[4] A. Sahai, "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (IEEE FOCS '99)*, pp. 543–553, IEEE, New York, NY, USA, October 1999.

[5] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," in *Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, Advances in Cryptology – EUROCRYPT '08*, N. Smart, Ed., vol. 4965 of *Lecture Notes in Computer Science*, pp. 415–432, Springer, Berlin, Germany, April 2008.

[6] M. Abe, K. Haralambiev, and M. Ohkubo, "Signing on elements in bilinear groups for modular protocol design," Cryptology ePrint Archive 2010/133, 2010, http://eprint.iacr.org/.

[7] B. Libert and M. Joye, "Group signatures with message-dependent opening in the standard model," in *Proceedings of the The Cryptographer's Track at the RSA Conference 2014, Topics in Cryptology – CT-RSA '14*, J. Benaloh, Ed., vol. 8366 of *Lecture Notes in Computer Science*, pp. 286–306, Springer International Publishing, San Francisco, CA, USA, February 2014.

[8] S.-H. Heng and K. Kurosawa, "k-resilient identity-based encryption in the standard model," in *Proceedings of the The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, T. Okamoto, Ed., vol. 2964, pp. 67–80, Springer, Berlin, Germany, 2004.

[9] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee, "Shorter IBE and Signatures via Asymmetric Pairings," in *Proceedings of the 5th International Conference, Cologne, Germany, May 16-18, 2012, Revised Selected Papers, Pairing-Based Cryptography – Pairing 2012*, M. Abdalla and T. Lange, Eds., vol. 7708 of *Lecture Notes in Computer Science*, pp. 122–140, Springer, Berlin, Germany, 2013.

[10] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proceedings of the 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings, Advances in Cryptology – CRYPTO 2004*, M. Franklin, Ed., vol. 3152 of *Lecture Notes in Computer Science*, pp. 41–55, Springer, Berlin, Germany, 2004.

[11] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, pp. 586–615, 2003.

[12] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM Journal on Computing*, vol. 33, no. 1, pp. 167–226, 2003.

[13] H. Shacham, "A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants," Cryptology ePrint Archive 2007/074, 2007, http://eprint.iacr.org/2007/074.

[14] J. Kilian and E. Petrank, "Identity escrow," in *Proceedings of the 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23-27, 1998, Proceedings, Advances in Cryptology – CRYPTO '98*, H. Krawczyk, Ed., vol. 1462, pp. 169–185 of *Lecture Notes in Computer Science*, pp. 169–185, Springer, Berlin, Germany, 1998.

[15] H. Arai, K. Emura, and T. Hayashi, "A framework of privacy preserving anomaly detection: providing traceability without big brother," in *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, pp. 111–122, ACM, 2017.

[16] G. Ateniese and G. Tsudik, "Some open issues and new directions in group signatures," in *Proceedings of the 3rd International Conference, FC '99 Anguilla, British West Indies, February 22-25, 1999, Proceedings, Financial Cryptography*, M. Franklin, Ed., vol. 1648 of *Lecture Notes in Computer Science*, pp. 196–211, Springer, Berlin, Germany, 1999.

[17] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proceedings of the 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings, Advances in Cryptology – CRYPTO 2004*, M. Franklin, Ed., vol. 3152 of *Lecture Notes in Computer Science*, pp. 56–72, Springer, Berlin, Germany, 2004.

[18] A. Kiayias and M. Yung, "Group signatures with efficient concurrent join," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, Advances in Cryptology – EUROCRYPT 2005*, R. Cramer, Ed., vol. 3494 of *Lecture Notes in Computer Science*, pp. 198–214, Springer, Berlin, Germany, 2005.

[19] J. Furukawa and H. Imai, "An efficient group signature scheme from bilinear maps," in *Proceedings of the 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings, Information Security and Privacy*, C. Boyd and J. M. G. Nieto, Eds., vol. 3574 of *Lecture Notes in Computer Science*, pp. 455–467, Springer, Berlin, Germany, 2005.

[20] C. Delerablée and D. Pointcheval, "Dynamic fully anonymous short group signatures," in *Proceedings of the 1st International Conference on Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers, Progress in Cryptology - VIETCRYPT 2006*, P. Q. Nguyen, Ed., vol. 4341, pp. 193–210, Springer, Berlin, Germany, 2006.

[21] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi, "Get shorty via group signatures without encryption," in *Proceedings of the 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010, Proceedings, Security and Cryptography for Networks*, J. A. Garay and R. De Prisco, Eds., vol. 6280 of *Lecture Notes in Computer Science*, pp. 381–398, Springer, Berlin, Germany, 2010.

[22] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Proceedings of the The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings, Topics in Cryptology - CT-RSA 2016*, K. Sako, Ed., vol. 9610 of *Lecture Notes in Computer Science*, pp. 111–126, Springer International Publishing, 2016.

[23] B. Libert, F. Mouhartem, T. Peters, and M. Yungy, "Practical "signatures with efficient protocols" from simple assumptions," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 511–522, ACM, June 2016.

[24] N. Barić and B. Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceedings, Advances in Cryptology – EUROCRYPT '97*, W. Fumy, Ed., vol. 1233, pp. 480–494, Springer, Berlin, Germany, 1997.

[25] E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations," in *Proceedings of the 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17-21, 1997, Proceedings, Advances in Cryptology – CRYPTO '97*, B. S. Kaliski Jr., Ed., vol. 1294 of *Lecture Notes in Computer Science*, pp. 16–30, Springer, Berlin, Germany, 1997.

[26] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Proceedings of the 6th Annual International Workshop, SAC '99, Kingston, Ontario, Canada, August 9-10, 1999, Proceedings, Selected Areas in Cryptography*, H. Heys and C. Adams, Eds., vol. 1758 of *Lecture Notes in Computer Science*, pp. 184–199, Springer, Berlin, Germany, 2000.

[27] D. Boneh and X. Boyen, "Short signatures without random oracles and the SDH assumption in bilinear groups," *Journal of Cryptology*, vol. 21, no. 2, pp. 149–177, 2008.

[28] G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros, "Practical group signatures without random oracles," Cryptology ePrint Archive 2005/385, 2005, http://eprint.iacr.org/.

[29] J. Groth, "Simulation-sound NIZK proofs for a practical language and constant size group signatures," in *Proceedings of the 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006, Proceedings, Advances in Cryptology – ASIACRYPT 2006*, X. Lai and K. Chen, Eds., vol. 4284 of *Lecture Notes in Computer Science*, pp. 444–459, Springer, Berlin, Germany, 2006.

[30] X. Boyen and B. Waters, "Compact group signatures without random oracles," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings, Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, pp. 427–444, Springer, Berlin, Germany, 2006.

[31] X. Boyen and B. Waters, "Full-domain subgroup hiding and constant-size group signatures," in *Proceedings of the 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings, Public Key Cryptography – PKC 2007*, T. Okamoto and X. Wang, Eds., vol. 4450 of *Lecture Notes in Comput. Sci.*, pp. 1–15, Springer, Berlin, Germany, 2007.

[32] J. Groth, "Fully anonymous group signatures without random oracles," in *Proceedings of the 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings, Advances in Cryptology – ASIACRYPT 2007*, K. Kurosawa, Ed., vol. 4833, pp. 164–180, Springer, Berlin, Germany, 2007.

[33] M. Bellare, H. Shi, and C. Zhang, "Foundations of group signatures: the case of dynamic groups," in *Proceedings of the The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings, Topics in Cryptology – CT-RSA 2005*, A. Menezes, Ed., vol. 3376 of *Lecture Notes in Computer Science*, pp. 136–153, Springer, Berlin, Germany, 2005.

[34] B. Libert, T. Peters, and M. Yung, "Short group signatures via structure-preserving signatures: standard model security from simple assumptions," in *Proceedings of the 35th Annual Cryptology Conference*, R. Gennaro and M. Robshaw, Eds., vol. 9216 of *Lecture Notes in Computer Science, Proceedings Part 2, Advances in Cryptology – CRYPTO 2015*, pp. 296–316, Springer, Santa Barbara, CA, USA, August 2015.

[35] J. Camenisch and M. Michels, "Separability and efficiency for generic group signature schemes," in *Proceedings of the 19th Annual International Cryptology Conference Santa Barbara, Advances in Cryptology – CRYPTO '99*, M. Wiener, Ed., vol. 1666 of *Lecture Notes in Computer Science*, pp. 413–430, Springer, California, USA, August 1999.

[36] A. Kiayias and M. Yung, "Group signatures: provable security, efficient constructions and anonymity from trapdoor-holders," Cryptology ePrint Archive 2004/076, 2004, http://eprint.iacr.org/.

[37] Y. Sakai, J. C. N. Schuldt, K. Emura, G. Hanaoka, and K. Ohta, "On the security of dynamic group signatures: preventing signature hijacking," in *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, Proceedings, Public Key Cryptography – PKC 2012*, M. Fischlin, J. Buchmann, and M. Manulis, Eds., vol. 7293 of *Lecture Notes in Computer Science*, pp. 715–732, Springer, Darmstadt, Germany, May 2012.

[38] B. Libert, T. Peters, and M. Yung, "Scalable group signatures with revocation," in *Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds., vol. 7237 of *Lecture Notes in Computer Science*, pp. 609–627, Springer, Cambridge, UK, April 2012.

[39] B. Libert, T. Peters, and M. Yung, "Group signatures with almost-for-free revocation," in *Proceedings of the 32nd Annual Cryptology Conference, Advances in Cryptology – CRYPTO 2012*, R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *Lecture Notes in Computer Science*, pp. 571–589, Springer, Santa Barbara, CA, USA, August 2012.

[40] N. Attrapadung, K. Emura, G. Hanaoka, and Y. Sakai, "A revocable group signature scheme from identity-based revocation techniques: achieving constant-size revocation list," in *Proceedings of the 12th International Conference, ACNS 2014, Applied Cryptography and Network Security*, I. Boureanu, P. Owesarski, and S. Vaudenay, Eds., vol. 8479 of *Lecture Notes in Computer Science*, pp. 419–437, Springer International Publishing, Lausanne, Switzerland, June 2014.

[41] T. Nakanishi and N. Funabiki, "Revocable group signatures with compact revocation list using accumulators," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98.A, pp. 117–131, 2015.

[42] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable signatures," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds., vol. 3027 of *Lecture Notes in Computer Science*, pp. 571–589, Springer, Interlaken, Switzerland, May 2004.

[43] E. J. Schwartz, D. Brumley, and J. M. McCune, "A contractual anonymity system," NDSS 2010. The Internet Society, 2010.

[44] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004*, pp. 168–177, ACM, USA, October 2004.

[45] Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote, "Group signatures with message-dependent opening," in *Proceedings of the 5th International Conference, Revised Selected Papers, Pairing-Based Cryptography – Pairing 2012*, M. Abdalla and T. Lange, Eds., vol. 7708 of *Lecture Notes in Computer Science*, pp. 270–294, Springer, Cologne, Germany, May 2012.

[46] K. Ohara, Y. Sakai, K. Emura, and G. Hanaoka, "A group signature scheme with unbounded message-dependent opening," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013*, pp. 517–522, ACM, May 2013.

[47] B. Libert, F. Mouhartem, and K. Nguyen, "A lattice-based group signature scheme with message-dependent opening," in *Proceedings of the 14th International Conference, ACNS 2016, Applied Cryptography and Network Security*, M. Manulis, A.-R. Sadeghi, and S. Schneider, Eds., vol. 9696 of *Lecture Notes in Computer Science*, pp. 137–155, Springer International Publishing, Guildford, UK, June 2016.

[48] H. Wee, "Public key encryption against related key attacks," in *Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, Proceedings, Public Key Cryptography – PKC 2012*, Fischlin., J. Buchmann, and M. Manulis, Eds., vol. 7293 of *Lecture Notes in Computer Science*, pp. 262–279, Springer, Darmstadt, Germany, May 2012.

[49] V. Shoup and R. Gennaro, "Securing threshold cryptosystems against chosen ciphertext attack," *Journal of Cryptology*, vol. 15, no. 2, pp. 75–96, 2002.

[50] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup, "Tag-KEM/DEM: a new framework for hybrid encryption and a new analysis of kurosawa-desmedt KEM," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT 2005*, R. Cramer, Ed., vol. 3494 of *Lecture Notes in Computer Science*, pp. 128–146, Springer, Aarhus, Denmark, May 2005.

[51] P. MacKenzie, M. K. Reiter, and K. Yang, "Alternatives to non-malleability: Definitions, constructions, and applications," in *Proceedings of the Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, M. Naor, Ed., vol. 2951 of *Lecture Notes in Computer Science*, pp. 171–190, Springer, Cambridge, MA, USA, February 2004.

[52] E. Kiltz, "Chosen-ciphertext security from tag-based encryption," in *Proceedings of the Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, S. Halevi and T. Rabin, Eds., vol. 3876 of *Lecture Notes in Computer Science*, pp. 581–600, Springer, New York, NY, USA, March 2006.

[53] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT 2003*, E. Biham, Ed., vol. 2656 of *Lecture Notes in Computer Science*, pp. 255–271, Springer, Warsaw, Poland, May 2003.

[54] M. Abe, K. Haralambiev, and M. Ohkubo, "Group to group commitments do not shrink," in *Proceedings of the 31st Annual*

International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT 2012, D. Pointcheval and T. Johansson, Eds., vol. 7237 of *Lecture Notes in Computer Science*, pp. 301–317, Springer, Cambridge, UK, April 2012.

[55] D. Boneh, P. A. Papakonstantinou, C. Rackoff, Y. Vahlis, and B. Waters, "On the impossibility of basing identity based encryption on trapdoor permutations," in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*, pp. 283–292, IEEE, October 2008.

[56] M. Abdalla and B. Warinschi, "On the minimal assumptions of group signature schemes," in *Proceedings of the 6th International Conference, ICICS 2004, Information and Communications Security*, J. Lopez, S. Qing, and E. Okamoto, Eds., vol. 3269 of *Lecture Notes in Computer Science*, pp. 1–13, Springer, Malaga, Spain, October 2004.

[57] G. Ohtake, A. Fujii, G. Hanaoka, and K. Ogawa, "On the theoretical gap between group signatures with and without unlinkability," in *Proceedings of the 2nd International Conference on Cryptology in Africa, Proceedings, Progress in Cryptology – AFRICACRYPT 2009*, B. Preneel, Ed., vol. 5580 of *Lecture Notes in Computer Science*, pp. 149–166, Springer, Gammarth, Tunisia, June 2009.

[58] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Advances in Cryptology – EUROCRYPT 2002*, L. R. Knudsen, Ed., vol. 2332 of *Lecture Notes in Computer Science*, pp. 65–82, Springer, Amsterdam, The Netherlands, April 2002.

[59] J. Camenisch, N. Chandran, and V. Shoup, "A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks," in *Proceedings of the 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Advances in Cryptology - EUROCRYPT 2009*, A. Joux, Ed., vol. 5479 of *Lecture Notes in Computer Science*, pp. 351–368, Springer, Cologne, Germany, April 2009.

[60] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo, "Structure-preserving signatures and commitments to group elements," in *Proceedings of the 30th Annual Cryptology Conference, Proceedings, Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, pp. 209–236, Springer, Santa Barbara, Calif, USA, August 2010.

[61] E. Ghadafi, N. P. Smart, and B. Warinschi, "Groth–Sahai proofs revisited," in *Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography, Proceedings, Public Key Cryptography – PKC 2010*, P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056 of *Lecture Notes in Computer Science*, pp. 177–192, Springer, Paris, France, May 2010.

[62] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[63] J. Furukawa and H. Imai, "An efficient group signature scheme from bilinear maps," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E89-A, no. 5, pp. 1328–1338, 2006.

[64] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki, "Revocable group signature schemes with constant costs for signing and verifying," in *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography, Proceedings, Public Key Cryptography – PKC 2009*, S. Jarecki and G. Tsudik, Eds., vol. 5443 of *Lecture Notes in Computer Science*, pp. 463–480, Springer, Irvine, Calif, USA, March 2009.

[65] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *Proceedings of the 18th Annual International Cryptology Conference, Proceedings, Advances in Cryptology – CRYPTO '98*, H. Krawczyk, Ed., vol. 1462 of *Lecture Notes in Computer Science*, pp. 13–25, Springer, Santa Barbara, Calif, USA, 1998.

[66] M. Bellare and G. Neven, "Multi-signatures in the plain public-Key model and a general forking lemma," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06*, pp. 390–399, ACM, 2006.

[67] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," Cryptology ePrint Archive 2004/332, 2004, http://eprint.iacr.org/2004/332.